

Grado en Ingeniería de Sistemas Audiovisuales

Trabajo Fin de Grado

“Desarrollo de una plataforma web de masterización adaptativa de ficheros de audio”

José Perona Jiménez

Tutor/es

David Griol Barres

Getafe, septiembre de 2018

RESUMEN

El auge de las nuevas tecnologías ha propiciado el avance de la producción musical en las últimas décadas. Gracias a ellas, productores independientes han conseguido construir sus propios estudios musicales destinando a ello una ínfima parte de lo habitual y consiguiendo unas prestaciones similares.

En este trabajo analizaremos una de las partes más importantes del proceso de creación de un tema musical en la actualidad, el mastering. Analizaremos la situación actual en la que se encuentra, recogeremos los problemas más importantes que surgen a la hora de realizarlo y propondremos la creación de una plataforma web que automatice todo el proceso, lo mejore mediante el *feedback* del usuario, y utilice las últimas tecnologías para conseguir un resultado óptimo.

Gracias al almacenamiento y el procesado de grandes volúmenes de datos, seremos capaces de crear diversos algoritmos que nos permitirán elegir los valores óptimos para que el resultado sea exitoso, etiquetando cada una de estas iteraciones para posteriormente ser capaces de entrenar un modelo de inteligencia artificial que agilice todo el proceso y sea capaz de aprender y mejorar con su uso.

A lo largo del documento también se explicará la arquitectura de esta solución modular, haciendo énfasis en la parte más técnica y de diseño, buscando un compromiso entre las prestaciones y la dificultad de desarrollo que nos permitan de forma sencilla completar el proyecto.

Palabras clave: Mastering, Inteligencia Artificial, Big Data, Plataforma Web, Música.

AGRADECIMIENTOS

Han pasado unos cuantos años desde que comencé mis estudios universitarios. Cuando aparecí los primeros días por la universidad, no sabía el enorme sacrificio que conllevaría dedicar una parte de tu vida a una ingeniería.

Hoy puedo decir con orgullo que cada uno de los retos que se pusieron ante mí los acabé superando. No fui un alumno brillante, no fui alguien destacado dentro de mi promoción y ni tan siquiera conseguí encajar en el modelo universitario, teniendo discrepancias varias durante todos mis estudios.

Pero he conseguido mi objetivo, hoy tengo un trabajo en el que dedico una gran parte de mi vida a la innovación tecnológica, a desarrollarme como persona con la tecnología y por el que me levanto por la mañana con la ilusión de conseguir nuevos retos. La universidad me ha aportado disciplina, madurez, saber que por muy imposible que se pongan las cosas tendrán solución, una filosofía de vida, y en eso tengo que estarle agradecido y olvidar los malos momentos, que ya quedaron atrás.

Gracias a mis padres, a mi hermano, a mi novia, por todo el apoyo que me han brindado estos años ya que si no fuera por todos ellos hoy no sería la persona que soy.

Gracias a mi tutor David por permitirme desarrollar un proyecto entero que tenía en mente desde hacía años y darme total libertad creativa y de diseño.

Gracias a todas las personas que han pasado en esta época por mi vida, ahora es el momento de cerrar esta página y comenzar un nuevo capítulo.

ÍNDICE GENERAL

RESUMEN	3
AGRADECIMIENTOS	5
ÍNDICE GENERAL	6
ÍNDICE DE FIGURAS	9
CAPÍTULO 1: INTRODUCCIÓN	11
1.1 Contexto	11
1.1.1 La música como registro sonoro	11
1.1.2 El proceso técnico de creación de un tema musical	12
1.1.3 La creación musical y las nuevas tecnologías	12
1.1.4 La transformación digital de las herramientas de edición	13
1.1.5 Los <i>Home studio</i>	14
1.1.6 El Mastering Digital	15
1.2 Motivación	16
1.3 Objetivos	17
1.4 Estructura del documento	18
CAPITULO 2: ESTADO DEL ARTE	20
2.1 El proceso actual de Mastering	20
2.1.1 Ecualización	21
2.1.2 Compresión	22
2.1.3 Limitación	23
2.2 Big data	24
2.3 Inteligencia Artificial en las aplicaciones web	25
2.4 Aplicaciones similares	27
2.5 Diferenciación del proyecto	28
CAPITULO 3: ANÁLISIS	30
3.1 Caso de uso y requisitos	30
3.2 Planteamiento por capas	30
3.2.1 Capa Front	32
3.2.2 Capa Back	33
3.3.3 Capa DataBase	34
3.3.4 Capa Big Data	34

3.3.5 Capa IA	35
3.3.6 Capa Processing	35
CAPÍTULO 4: DISEÑO	37
4.1 Arquitectura general de la solución	37
4.2 Diseño Front	38
4.2.1 Segmento de presentación	39
4.2.2 Segmento funcionamiento	39
4.2.3 Segmento estadísticas	40
4.2.4 Segmento de contacto	40
4.2.5 Segmento subir Archivo	41
4.2.6 Lógica de la página	41
4.2.7 Tecnologías utilizadas	42
4.3 Diseño Back	43
4.3.1 Routing	45
4.3.2 Conexión con la base de datos	46
4.4 Diseño Database	47
4.4.1 Arquitectura de la base de datos	48
4.4.2 Modelo de datos de las colecciones	49
4.5 Diseño Procesado	52
4.5.1 Micro servicio de mastering	52
4.5.2 Routing	53
4.5.3 Herramientas de mastering	54
4.6 Diseño Big Data	57
4.6.1 Lógica de tratamiento de los datos	57
4.6.2 Visualización y analytics	59
4.7 Diseño IA	60
4.7.1 Obtención del dataset	61
4.7.2 Entrenamiento del modelo	61
4.7.3 Inferencia	62
CAPITULO 5: CONCLUSIONES Y TRABAJO FUTURO	64
5.1 Trabajo Futuro	65
CAPÍTULO 6: GESTIÓN DEL PROYECTO	67
6.1 Planificación del proyecto	67
6.2 Presupuesto	68
6.3 Impacto socio-económico	69

6.4 Marco Regulador	70
ENGLISH APPENDIX	73
GLOSARIO	78
SIGLAS, ACRÓNIMOS Y ABREVIATURAS	80
BIBLIOGRAFÍA	81

ÍNDICE DE FIGURAS

<i>Figura 1. Fairchild modelo 660 de 20kg de peso</i>	13
<i>Figura 2. Ejemplo de home studio</i>	14
<i>Figura 3. Mastering Chain</i>	21
<i>Figura 4. Ecualizador paramétrico digital</i>	22
<i>Figura 5. Compresor digital HCOMP de Waves</i>	23
<i>Figura 6. Limitador Digital L2 de Waves.</i>	24
<i>Figura 7. Peso económico del Big Data [8]</i>	25
<i>Figura 8. Booms de la IA a través de los años</i>	26
<i>Figura 9. Estructura por capas del proyecto</i>	31
<i>Figura 10. Arquitectura general de la solución.</i>	37
<i>Figura 11. Segmento “conócenos”</i>	39
<i>Figura 12. Segmento “como funciona”</i>	39
<i>Figura 13. Segmento “contador”</i>	40
<i>Figura 14. Segmento “contáctanos”</i>	40
<i>Figura 15. Vista inicial de la web “subir archivo”</i>	41
<i>Figura 16. : Zoom al módulo front</i>	42
<i>Figura 17. Lenguajes de programación Back-End más populares [13]</i>	43
<i>Figura 18. Focus de los desarrollos en NodeJS [14]</i>	44
<i>Figura 19. Arquitectura Back-end en detalle</i>	45
<i>Figura 20. Modelo de datos JSON de la capa Back-end</i>	46
<i>Figura 21. Arquitectura de la base de datos</i>	48
<i>Figura 22. Modelo de datos de la colección Pre/Post-features</i>	49
<i>Figura 23. Modelo de datos de la colección Mastering Parameters</i>	50
<i>Figura 24. Modelo de datos de la colección Analitical Data</i>	51
<i>Figura 25. Arquitectura del micro servicio de mastering</i>	53
<i>Figura 26. Division por bandas del espectro audible</i>	54
<i>Figura 27. Ejemplo de filtro de pico en Python centrado en 20Khz.</i>	55
<i>Figura 28. Audio antes y después de la limitación</i>	56
<i>Figura 29. Ejemplo de visualización de datos en Compass</i>	60
<i>Figura 30. Ejemplo de red neuronal</i>	62
<i>Figura 31. Arquitectura final con el módulo de IA</i>	63

<i>Figura 32. Diagrama Gantt del proyecto</i>	67
<i>Figura 33. Presupuesto del proyecto</i>	68

CAPÍTULO 1: INTRODUCCIÓN

En este documento se describe el proceso de realización del trabajo final de grado consistente en la realización de una plataforma web de masterización adaptativa de ficheros de audio.

En este capítulo explicaremos la relevancia de la creación de esta plataforma, utilizando para ello el contexto que nos ha guiado hasta ésta solución en primer lugar, las motivaciones pertinentes para su consecución y los objetivos definidos para obtener un resultado final óptimo y fiable.

Por último, mostraremos la estructura general organizada en capítulos y anexos para la correcta lectura y entendimiento del documento.

1.1 Contexto

1.1.1 La música como registro sonoro

En 1877 una de las mentes más brillantes de la humanidad, Thomas Alva Edison, concibió el sonido como algo capaz de ser registrado y posteriormente reproducido. La invención del fonógrafo fue uno de los mayores hitos de la historia musical, frecuentemente olvidado por la aparición de su hermano mayor el gramófono [1].

Más de 140 años después, los avances conseguidos en este ámbito han sido innumerables, pasando por la creación del disco de vinilo, el magnetófono de bobina abierta, el casete compacto, el cd, etc. Avances que han permitido hacer perdurar las obras musicales en diversos formatos, y no únicamente como partituras interpretables.

A medida que la tecnología avanzaba, la música y ella se daban la mano para conseguir cualidades técnicas que permitieran una escucha agradable y portable. Es esa portabilidad, la que hoy nos permite disfrutar de la música en cualquier parte gracias a los dispositivos móviles, con una calidad excepcional.

1.1.2 El proceso técnico de creación de un tema musical

Si bien la grabación del sonido es el aspecto básico y fundamental de la elaboración de un tema musical, es la suma de estas grabaciones y su posterior edición técnica la responsable de la calidad final de éste. Así pues, podremos dividir el proceso técnico de creación de un tema en tres partes diferenciadas entre sí: grabación, mezcla y mastering:

- **Grabación:** Es la inscripción eléctrica o mecánica y la recreación de ondas sonoras. Sus dos clases principales se dividen en grabación analógica (a través de micrófonos generalmente) y grabación digital (con un uso extendido de sintetizadores de sonido).
- **Mezcla:** Es un proceso utilizado en la edición del sonido, mediante el cual se balancea y equilibra el volumen relativo y la ecualización de las fuentes de sonido que se encuentran presentes en un evento sonoro. La mezcla del audio puede ser muy relativa, ya que el mismo conjunto de sonidos mezclados por un ingeniero de sonido diferente puede dar lugar a obras muy diferenciadas entre sí que, a pesar de partir de la misma base, técnicamente y auditivamente distan mucho unas de las otras.
- **Mastering:** Es el proceso mediante el cual se adecúa una grabación sonora para ligarla a un producto final, del cual partirán las copias que luego serán distribuidas. Es el último paso en la elaboración de un tema musical, que asegura su calidad óptima mediante diferentes técnicas de procesado del audio y donde se corrigen los últimos errores.

Si bien la elaboración de un tema musical no tiene un guion concreto en lo que se refiere al apartado técnico, esta cadena de creación es la más utilizada por los diversos profesionales del sector [2].

1.1.3 La creación musical y las nuevas tecnologías

En los años 60, las herramientas necesarias para la creación de un tema musical podrían llegar a ocupar el tamaño de habitaciones (ver Figura 1). Compresores, ecualizadores, limitadores analógicos cuyo coste hacía que solo un pequeño grupo de personas pudiese acceder a esta tecnología y que se agrupaban en grandes estudios musicales donde se llevaba a cabo todo el proceso de grabación, mezcla y mastering.



Figura 1. Fairchild modelo 660 de 20kg de peso

Con el paso del tiempo y el avance de las nuevas tecnologías, estas herramientas se han ido desarrollando y abaratando su precio hasta tal punto que cualquier persona hoy en día podría montarse un estudio de grabación en su propia habitación, pudiendo obtener una calidad profesional sin necesidad de acudir a uno de estos grandes estudios.

El abaratamiento de este tipo de productos y el conocimiento compartido en la red, han sido el campo de cultivo perfecto para la proliferación de productores musicales independientes, en los que los llamados *home studios* han sido los principales protagonistas reemplazando poco a poco a los grandes estudios [3] [4].

1.1.4 La transformación digital de las herramientas de edición

Uno de los ingredientes principales de ese caldo de cultivo que hemos mencionado antes, ha sido sin ninguna duda la transformación digital de las herramientas de producción. El uso de secuenciadores digitales multipista, sintetizadores digitales de sonido y librerías de sonido digital ha sido uno de los grandes avances en este campo.

Una de las librerías de *plugins* más conocidas, *Waves*, lanzó en 1992 el famoso *plugin Q10* de ecualización paramétrica. Fue el primer *plugin* de audio comercial, que dio paso a toda una serie de *plugins* tanto de ecualización como de compresión. Hoy en día, *Waves* cuenta con una amplia gama de más de 200 productos a un precio asequible, actualizados constantemente y con un funcionamiento muy similar a su versión analógica.

Las mesas de mezclas analógicas han sido sustituidas poco a poco por secuenciadores digitales como *Protools*, quitando al usuario la necesidad de tener que destinar un

espacio a tener este tipo de elementos y reduciendo considerablemente el material necesario a adquirir para poder crear música.

1.1.5 Los *Home studio*

Llegamos a un punto clave del contexto, y es que con la ayuda de un pc, unos monitores de sonido y el software necesario, cualquier persona hoy en día puede crear piezas musicales con una calidad profesional.

Este entorno de trabajo, que suele constar también de una sala acondicionada acústicamente, permite que en muy poco espacio y por un módico precio se pueda realizar una grabación y edición de sonido que hace 20 años hubiesen requerido grandes estudios, varios ingenieros de sonido y una cantidad de componentes analógicos (desde instrumentos y micrófonos hasta compresores y ecualizadores) que a día de hoy serían inviables por el presupuesto económico que conllevarían, podemos observar así en la Figura 2, un *home studio* típico:



Figura 2. Ejemplo de home studio

Gracias a estos avances, la barrera de entrada al sector de la edición musical ha descendido de forma considerable, propiciando la aparición de estos productores musicales independientes. Una de las características que poseen este tipo de productores musicales es el conocimiento compartido, es decir, formarse en temas de edición y postproducción de audio normalmente requería años de práctica con un ingeniero que te transmitía tanto sus conocimientos como su forma de realizar los proyectos, pero con

esta inteligencia compartida y esta accesibilidad al software, han surgido recientemente infinidad de blogs con guías y demostraciones en directo sobre la utilización de esas herramientas para conseguir un sonido profesional.

Es en este punto donde haremos un inciso, porque si bien existen muchos focos de información sobre la utilización de este tipo de *plugins*, hay un paso en la cadena de la creación musical en que el “cómo” se usan estos plugins es mucho más relevante que el “cuando” se usan, y de ello pasaremos a hablar en el siguiente apartado.

1.1.6 El Mastering Digital

El mastering es el último eslabón de la cadena, el proceso más complejo donde la experiencia y el conocimiento del ingeniero de sonido tiene absoluta relevancia. La tecnología nos brinda los medios necesarios para poder tratar el sonido, pero el conocimiento que hay que aplicar es muy amplio y solo una fracción pequeña de las personas que se dedican a la profesión posee conocimientos específicos sobre mastering.

Así, a la hora de producir un tema musical, lo más probable es que una vez grabado y mezclado ese archivo se envíe a un ingeniero de mastering para que de los últimos retoques y corrija errores, haciendo que el tema suene profesional y adaptándolo al estilo propio de la banda. No es extraño entonces pensar que un mismo ingeniero suele masterizar siempre a los mismos grupos, y aunque use técnicas similares en cada uno de sus proyectos los resultados serán muy variables entre sí.

En el Mastering se une tanto el proceso creativo como el proceso técnico, siendo este último el más importante debido a que los pasos a seguir para realizarlo correctamente están muy definidos, quedando poco margen abierto a la interpretación ya que nos encontramos con algo puramente técnico.

Hay que destacar en este punto, que debido a la alta complejidad de este proceso (como ya hemos mencionado, hay una fracción muy pequeña de gente que sabe hacerlo bien), los masterings suelen ser caros, un buen porcentaje económico del presupuesto para el desarrollo de un tema musical suele perderse en este proceso ya que hay que disponer durante horas de un ingeniero experimentado que consiga masterizar el tema de forma óptima, siendo ésta su única dedicación.

1.2 Motivación

Uno de los problemas que ya se han identificado anteriormente es lo que ha motivado la elaboración de este proyecto. En este caso, la dificultad que conlleva la realización del proceso de mastering, el coste, y el tiempo que hay que invertir en él hace que pensemos que puede existir una forma más rápida, eficaz y viable de realizar este tipo de procesados que tradicionalmente han sido manuales.

El avance de las herramientas de procesamiento (*plugins*) permite la elaboración de diversos algoritmos para el tratamiento del sonido que, unidos en una cadena, pueden simular el tratamiento del audio que se da en el mastering. Si somos capaces de analizar los procesos a usar dentro de este apartado también seremos capaces de automatizar este sistema, creando una herramienta de mastering automático que nos permita replicar esa cadena sin ningún esfuerzo y configurando “únicamente” una serie de variables.

Además, el auge de las plataformas web nos da la opción de crear una instancia en la que integrar este servicio, que nos dará diversos beneficios como:

- Acceso a una cantidad mucho mayor de personas que utilizarán nuestro servicio.
- Potencial escalado del proyecto.
- Capacidad de entrenar un algoritmo que pueda aprender de cada proceso de mastering realizado.
- Creación de una base de datos con todos y cada uno de los temas musicales analizados, junto con cada uno de sus parámetros técnicos.
- Posibilidad de adaptar el proceso de mastering a un estilo determinado por el usuario.
- Posibilidad de monetizar la plataforma web a través de publicidad o donaciones para el uso gratuito de la aplicación.

Así, podremos crear una plataforma web a la que cualquiera tenga acceso, que funcione gratuitamente y de la que podamos obtener modelos de procesamiento de audio que nos permitan afinar nuestra cadena de mastering, adecuándola también a los gustos de la persona que la está utilizando en cada momento.

Con esta aplicación intentamos poner una solución si bien no perfecta, rápida y eficaz para que pueda ser usada por los productores independientes que no tienen la capacidad económica para financiar un mastering, pero que buscan un resultado final en sus

trabajos de calidad profesional, o al menos, obtener una idea aproximada de cuál sería este resultado si se enviase a un estudio de masterización.

1.3 Objetivos

La finalidad de este proyecto consiste en la elaboración de una página web en formato *One-page*, a través de la que el usuario pueda navegar y subir sus archivos de audio para que sean procesados, descargando una versión masterizada como resultado final. La aplicación debe ser capaz de, una vez recogido el archivo de audio que el usuario sube, procesar el mismo en base a los parámetros definidos por nuestro algoritmo de masterización, contando para ello con diversas herramientas.

Por tanto, se crearán una serie de algoritmos y *scripts* que nos permitirán:

- Subir un archivo de audio en formato *wav* o *mp3* a través de la web y enviarlo en un *JSON* hacia un servidor que se encargue del procesado.
- Extraer las características principales del archivo de audio (Densidad espectral de potencia dividida por bandas, *bitrate*, tempo, etc.) y almacenarlas en una base de datos antes del procesado.
- Comparar las características del tema musical subido por el usuario con una base de datos con características de los procesos realizados para obtener temas profesionales y cuya calidad está contrastada, o con temas o estilos que el usuario haya seleccionado por defecto.
- En base a los resultados de la comparación, adaptar los diferentes valores de entrada de la cadena de Mastering.
- Crear una cadena de Mastering que en función de los parámetros de entrada, nos devuelva un audio masterizado a la salida con una calidad lo más cercana posible a un archivo profesional, utilizando para ello herramientas de equalización, compresión y limitación del audio.
- Extraer las características finales del tema procesado y almacenarlas en otra base de datos.
- Crear una capa de análisis de los procesos realizados, en la que comparemos los archivos antes del proceso y después de éste, obteniendo información en tiempo real sobre qué ha ocurrido en cada uno de los procesos y cómo ha funcionado el algoritmo, identificando los estilos musicales o temas que se han utilizado y el

usuario que lo ha llevado a cabo, y extrayendo un perfil completo individual de cada uno de los usuarios.

- Obtener el *feedback* del usuario
- Crear una inteligencia artificial que aprenda sobre los procesos exitosos de mastering, adaptándose a cada usuario tras su uso, y creando una experiencia de aprendizaje y optimización del proceso.
- La descarga sencilla de un archivo final.

Por tanto, tendremos una herramienta muy completa que nos soluciona en gran medida un problema, que a su vez nos sirve como proveedora de diferentes datos que serán recogidos de los propios usuarios para su posterior uso en la mejora y optimización de los algoritmos, dándonos también la opción de orientar estos datos a un enfoque económico como puede ser las recomendaciones en función del perfil musical de cada uno de los usuarios, y todo ello de forma gratuita para que esos productores independientes puedan acceder a un tipo de servicio que suele estar restringido a las personas con más dinero o proyectos más sólidos.

1.4 Estructura del documento

La estructura de la memoria es la siguiente:

Capítulo 1: Introducción. Realizaremos un recorrido sobre la historia que ha dado lugar a los diferentes procesos de audio, cómo ha influido la tecnología en estos y las motivaciones y objetivos de nuestro proyecto

Capítulo 2: Estado del arte. Mostraremos el estado actual del proceso de masterización y qué elementos lo componen, además de analizar el estado general de las diferentes tecnologías que utilizaremos en nuestro proyecto. También hablaremos de las aplicaciones similares que se encuentran en la web y la diferenciación de nuestro proyecto.

Capítulo 3: Análisis. Plantearemos el problema mediante un análisis por capas para la correcta identificación de cada uno de los requisitos necesarios para la consecución del proyecto.

Capítulo 4: Diseño. Crearemos una plataforma web de mastering adaptativo a través del diseño y el desarrollo de diversos módulos y micro servicios, explicando cada una de las tecnologías utilizadas y el motivo de su elección.

Capítulo 5: Conclusiones y trabajo futuro. En este capítulo expondremos las principales conclusiones a las que hemos llegado tras la realización del proyecto, incluyendo el trabajo a realizar tras la presentación.

Capítulo 6: Gestión del proyecto. En este capítulo hablaremos sobre los tiempos del proyecto, el presupuesto que se ha elaborado y su posible impacto en el sector. También hablaremos sobre la actual legislación aplicable a nuestro proyecto en relación con la sociedad general de autores y el actual reglamento general de protección de datos.

CAPITULO 2: ESTADO DEL ARTE

Cuando investigamos sobre el proceso de mastering, llegamos a la conclusión de que prácticamente la totalidad de los masterings profesionales se producen en unas condiciones muy similares: se hacen de forma manual, necesitan la intervención durante horas de un ingeniero de sonido experimentado que tenga conocimientos específicos sobre este campo, se producen en grandes estudios de sonido acondicionados acústicamente y requieren de un desembolso enorme en herramientas analógicas. Por ello vamos a explicar cómo se realiza un mastering a día de hoy, y en qué estado se encuentran las tecnologías que vamos a utilizar para llevar a cabo tanto nuestra plataforma web como el *core* de nuestra aplicación [5].

2.1 El proceso actual de Mastering

Si bien el proceso de Mastering es algo complejo que no tiene una estructura definida, sí que podemos encontrar un patrón que siguen la mayoría de productores a la hora de crear su propia *Mastering Chain*, o consecución de plugins para tratar el sonido para obtener un resultado final óptimo, en el cual podemos encontrar tres elementos básicos como son la ecualización, la compresión y la limitación, y otros plugins para retocar armónicos, compresores multibanda, etc.

Así, la cadena que un ingeniero suele usar es la siguiente:

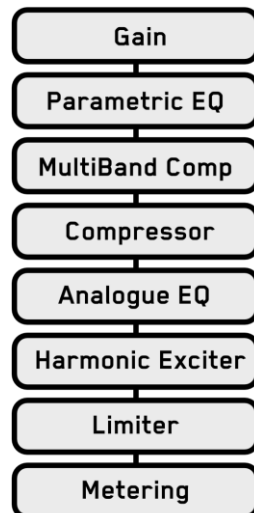


Figura 3. Mastering Chain

Como podemos observar en la Figura 3, tenemos un primer módulo compuesto por una ecualización paramétrica, un segundo módulo compuesto por compresores, y un módulo final en el que limitamos nuestro archivo de audio.

2.1.1 Ecualización

A la hora de ecualizar, nuestras tareas como ingenieros de sonido serán varias:

- Eliminar resonancias y corregir pequeños errores en la mezcla.
- Ajustar la relación entre graves, medios y agudos.
- Dar una amplitud estéreo cuando sea necesario.
- Resaltar las frecuencias características de nuestro tema musical, dependiendo de cuál sea nuestro estilo/género.

Para llevar a cabo una correcta ecualización los productores de sonido se valen de los ecualizadores paramétricos (ver Figura 4), que no solo nos permiten dividir el espectro en diferentes bandas de frecuencia cuyas ganancias podemos modificar, sino que también nos permiten ajustar el factor “Q” para tener mucha mayor precisión a la hora de modificar estas bandas.



Figura 4. Ecualizador paramétrico digital

Los ecualizadores digitales actuales suelen imitar las características de sus homólogos analógicos, por eso distinguir en un producto final cuando se ha usado uno u otro suele ser muy complicado. Uno de los mejores desarrolladores de este tipo de plugins es la empresa Waves, que posee una suite de herramientas específicas para el masterizado, dentro de la que se encuentran diversos ecualizadores paramétricos como pueden ser los ecualizadores Q10, SSL y V-EQ4, por citar algunos ejemplos.

2.1.2 Compresión

Siguiendo con la cadena de Mastering que detallamos en el primer apartado, una vez ecualizado nuestro tema pasaremos a retocar su rango dinámico, mediante la utilización de compresores (ver Figura 5). En este paso, los objetivos a alcanzar son los siguientes:

- Añadir color a la mezcla.
- Comprimir picos de sonido que se hayan producido como resultado de una mala mezcla.
- Controlar los niveles de graves y agudos para que podamos identificarlos correctamente y no se superpongan si trabajamos con un compresor multibanda.



Figura 5. Compresor digital HCOMP de Waves

Si bien es cierto que la compresión no tiene un papel tan relevante como la ecualización, esto es debido a que posteriormente en la cadena de mastering se realizará una limitación del archivo de audio, siendo la limitación una forma de compresión con unos ratios específicos.

2.1.3 Limitación

El último gran bloque de la cadena de Mastering y el que más influencia tendrá en la percepción de la calidad del sonido final será la limitación, que será realizada con compresores específicos que contendrán parámetros seleccionados para su uso como limitador (ver Figura 6).

La limitación en el mastering consiste en la compresión agresiva de los picos que superan un cierto umbral y en conseguir un volumen final elevado a través de los parámetros establecidos por la persona que configura el compresor. Lo que se traduce en obtener una cantidad de RMS promedio más elevada, dando así una sensación de sonoridad mayor.

Para ejemplificar la importancia de este proceso dentro de la cadena de mastering, es importante hacer mención a un término acuñado por la industria musical durante los últimos años llamado “la guerra del volumen” [6].

La guerra del volumen, es básicamente el crecimiento exponencial de los RMS promedio de los temas musicales desde los últimos años hasta hoy día. Esto es debido a fenómenos psicoacústicos que nos dicen que cuanto más fuerte suena un tema musical,

más valoración positiva subjetiva tendremos de él, por lo que cuanto más limitemos un tema sin llegar a perder su dinámica u obtener distorsión, mejor percepción de él tendrán los oyentes.



Figura 6. Limitador Digital L2 de Waves.

2.2 Big data

Una de las principales cualidades que deseamos de nuestra aplicación y por la cual será diferencial es el poder afinar el proceso lo suficiente como para que cualquier persona pueda hacer uso de él obteniendo unos resultados satisfactorios. Para llevarlo a cabo, necesitaremos una cantidad de datos catalogados enorme, que nos permita mediante el aprendizaje extraer una serie de patrones de aquellas obras que consideramos de una calidad excelente y almacenarlas para su posterior uso.

El tratamiento de estas cantidades tan elevadas de datos es a lo que denominamos hoy en día “Big Data”, una forma de tratar los datos que nos permite el desarrollo de aplicaciones más eficientes, escalables y complejas, permitiéndonos llevar el análisis de nuestros datos a nuevos niveles para conseguir por tanto mejores aplicaciones. Para poder comprobar la relevancia de esta tecnología en la actualidad, podemos observar en la Figura 7 como el peso económico en miles de millones de dólares de inversión en esta tecnología hace prever que nos encontramos en un punto donde el Big Data se ha vuelto casi fundamental dentro de las empresas y aplicaciones que trabajan con grandes volúmenes de datos [7].

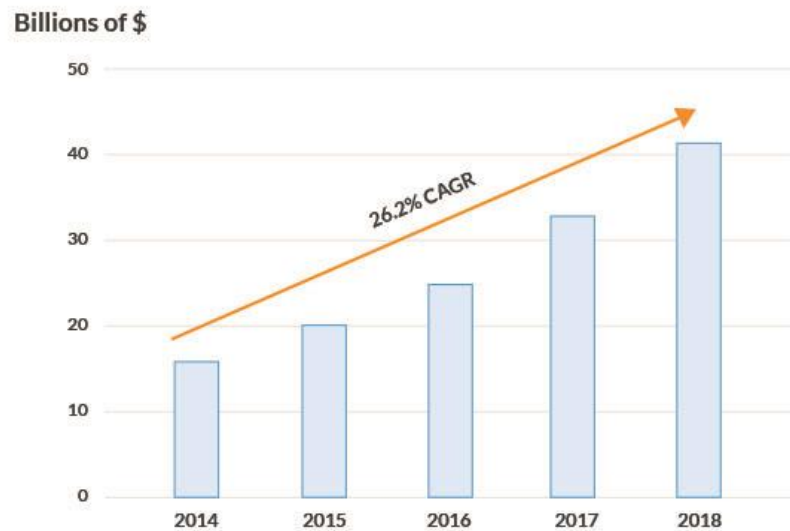


Figura 7. Peso económico del Big Data [8]

Así, hemos decidido adoptar esta tecnología debido a su enorme potencial y a que en nuestro proyecto sí que se dan las condiciones necesarias para su uso. Si bien en este desarrollo no contaremos con tal cantidad de datos, sí que se planteará una arquitectura escalable que nos permita comprobar de forma práctica los posibles casos de uso que se plantean.

2.3 Inteligencia Artificial en las aplicaciones web

Hablar de inteligencia artificial hoy en día es algo complejo, si bien nos encontramos ante una tecnología que ya existe desde hace años, no es hasta las últimas evoluciones en computación y la era *machine learning* cuando se está empezando a vislumbrar su potencial.

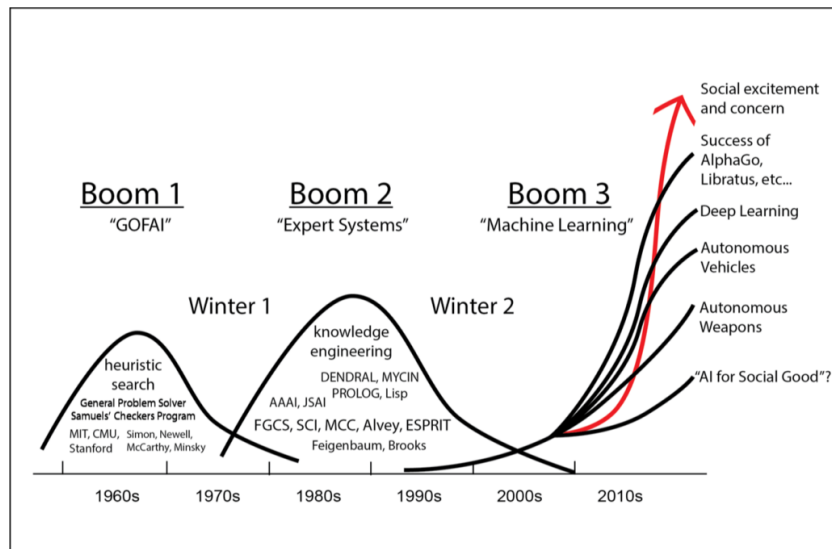


Figura 8. Booms de la IA a través de los años

Las diferentes aplicaciones web utilizan la inteligencia artificial de diversas formas, aunque es con ésta última variable citada, el *machine learning*, con la que han conseguido empezar a explotar las cualidades de ésta tecnología (ver Figura 8). Así, mediante la generación de un conjunto de datos etiquetados, consiguen entrenar una variedad de modelos que sirven para automatizar y optimizar cualquier proceso de la misma forma que un humano podría llevarlo a cabo.

En lo referente a herramientas relacionadas con IA, nos encontramos situada en el podio a *Tensorflow*, una biblioteca de código abierto desarrollada por Google para trabajar más fácilmente con el conjunto de algoritmos que componen el mundo de la IA, quitando la necesidad de ser un experto en la materia para poder desarrollar aplicaciones capaces de aprender de sus usuarios.

En este apartado podemos poner varios ejemplos sobre casos de éxito de la aplicación de esta tecnología a las plataformas web, como es el caso de Netflix. Esta empresa ha sido capaz de recoger una cantidad ingente de datos de sus usuarios, refinarlos, y entrenar modelos de recomendación de películas que consiguen que el usuario se mantenga dentro de la aplicación a la vez que se diferencian del resto de compañías que ofrecen exclusivamente un servicio “normal” de *streaming*.

También tenemos el caso de Amazon, que ha llevado su departamento de inteligencia artificial a un nivel superior, no solo creando sistemas recomendadores personalizados al usuario, si no también lanzando “*Alexa*”, un aparato con un micrófono y un altavoz

destinados al hogar capaz de comunicarse con una inteligencia que puede hacer de asistente personal.

2.4 Aplicaciones similares

Realizando una simple búsqueda en Google con el parámetro “mastering online”, podemos encontrar diversas plataformas web que ofrecen este tipo de servicios de mastering a un bajo coste, de todas ellas podemos destacar:

- **LANDR:** Una plataforma que no solo permite masterizar online el tema musical sino que también ofrece una serie de herramientas para poder distribuir tu música y generar una especie de red social destinada a productores de sonido, siendo capaz de valorar los temas musicales de otras personas y exponiendo los tuyos propios a un módico precio. Las ventajas que nos da esta plataforma son un *all-in-one* con el producto final, dándonos las guías para poder tener un producto con una calidad profesional e intentando captar al usuario para seguir el resto del camino de distribución de la música con ellos [9].
- **BandLab:** Una *startup* ganadora del premio “[Start-up Partner of the Year Award 2016](#)” [10]. Ofrece una plataforma de mastering gratuita, en la que destacan tres cualidades principales: una calidad de cd, mejora del sonido de los bajos y una claridad de sonido mejorada.
- **Emastered:** Esta empresa está posicionada dentro de la industria como la poseedora del mejor motor de mastering, gracias a que ha sido desarrollado junto con ingenieros de sonido ganadores de premios Grammy. Si bien es algo más cara que sus competidoras, ofrece una calidad excelente que puede servir a productores de sonido noveles como punto de entrada al sonido profesional [11].

Todas estas aplicaciones han sido creadas con la finalidad de automatizar el proceso de mastering y hacerlo accesible a la mayor cantidad de público posible mediante sus servicios web. Si bien todas ellas ofrecen un servicio con una calidad aceptable, los

procesos que usan son muy similares, pidiendo al usuario subir un archivo en un formato determinado con las siguientes características:

- Entre 3 dB y 6 dB de margen dinámico en el *headroom*.
- No utilizar ningún efecto en el canal máster durante la mezcla.
- Utilizar una profundidad de bits elevada para su mejor procesado.

Existe un problema intrínseco en todas ellas que servirá como base para el desarrollo de nuestra aplicación, aplicando un enfoque diferente al problema para obtener unos resultados diferentes, como explicaremos en el siguiente apartado.

2.5 Diferenciación del proyecto

Como hemos mencionado antes, existe un problema común dentro de todo este tipo de aplicaciones y es que sus masterings **no son adaptativos**. Los algoritmos que utilizan han sido diseñados por ingenieros de sonido con muchos años de experiencia que simplemente se limitan a aplicar una serie de filtros y compresiones como si de una regla se tratase, obviando el aprendizaje que se puede extraer de cada uno de estos procesos y obviando el feedback del usuario.

Si tras analizar la pista y concluir que debe ser tratada de una determinada manera y con unos parámetros obtenidos como resultado, no somos capaces de saber si nuestro algoritmo funciona adecuadamente o no, estamos perdiendo la oportunidad de personalizar todo el proceso y de depurar nuestro algoritmo para conseguir un mejor funcionamiento.

Así, gracias al almacenamiento de las características principales de un tema listo para ser procesado, los parámetros que hemos obtenido para utilizar en el algoritmo de mastering y el feedback del usuario, somos capaces de establecer una gran base de datos sobre la que podremos aplicar procesos Big Data que establezcan una relación entre los temas musicales masterizados correctamente, qué parámetros se han utilizado para obtener ese resultado exitoso y qué parámetros serán necesarios para obtener una probabilidad de éxito mayor a la hora de realizar este proceso cuando nos encontremos ante un nuevo tema musical, pudiendo consultar los pasos seguidos en cada uno de los procesos en tiempo real a través de herramientas de *data analytics*.

Toda esta cantidad de datos puede ser definida como un gran *dataset* de elementos etiquetados que nos permitirán entrenar un modelo de inteligencia artificial, mediante el cual optimizaremos todos los procesos obteniendo un mastering más rápido y de mejor calidad.

CAPITULO 3: ANÁLISIS

En este capítulo procederemos a explicar el proceso técnico de análisis seguido para la elaboración del proyecto comenzando por la toma de requisitos y exposición del caso principal de uso.

3.1 Caso de uso y requisitos

Nuestro caso de uso principal será la masterización de un archivo de audio subido por un usuario para su posterior descarga.

Una de las mayores complejidades de este proyecto se encuentra en el número de tecnologías utilizadas en él, por lo que al tener un elevado número de requisitos dependiendo del paso del mastering en el que nos encontremos, hemos decidido plantear un análisis por capas, en el que iremos desarrollando los requisitos necesarios para cada una de ellas, facilitando así el entendimiento del problema y el posterior diseño de una solución modular adecuada a él.

3.2 Planteamiento por capas

Así, dispondremos de 6 capas en nuestra estructura, cada una realizará una función diferente y todas estarán conectadas de tal forma que podamos crear nuestra plataforma de mastering adaptativo.

A continuación, podemos observar el planteamiento propuesto y una breve explicación sobre la funcionalidad de cada capa, pasando después a puntualizar cada una de ellas y realizar un análisis más exhaustivo sobre los requisitos necesarios para conseguir nuestros objetivos.

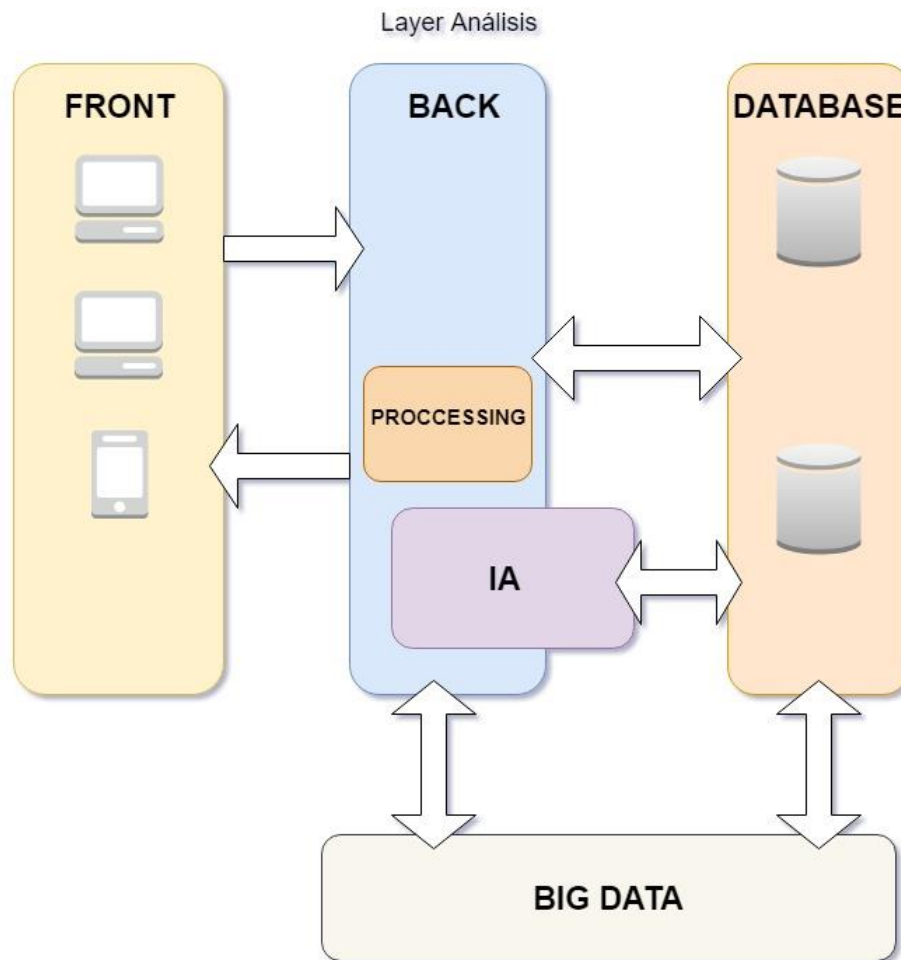


Figura 9. Estructura por capas del proyecto

Como podemos observar en la Figura 9, encontramos 6 capas diferentes:

- **Front layer:** Será la capa frontal de nuestra aplicación, una página web cuyo principal objetivo será constituir un punto de acceso a la aplicación de mastering y ser la referencia visual de todo el proyecto, permitiéndonos subir un archivo de audio y descargar el archivo final masterizado.
- **Back layer:** Aquí se producirán todas las conexiones con el resto de capas, será la capa encargada enviar el archivo para extraer sus características, recibirlas y almacenarlas, conectarse con la capa *Big Data* para saber qué parámetros se deben usar en el mastering, enviar esos parámetros y almacenar las características finales obtenidas en una base de datos diferente. También será la encargada de alimentar la conexión para la elaboración del modelo de inteligencia artificial que nos permitirá la optimización de todo el proceso.

- **Database layer:** será la capa encargada de almacenar todos los datos que se manejarán durante el proceso.
- **Big data layer:** Será la capa de tratamiento de los datos y características almacenadas, a través de la cual obtendremos unos parámetros de salida que nos permitirán masterizar nuestro tema adecuadamente. También contaremos con una herramienta de visualización de cada uno de los procesos en tiempo real y de análisis y explotación de todos los datos proporcionados por el usuario.
- **IA layer:** Será la capa encargada de entrenar y utilizar un modelo de inteligencia artificial para la mejora de la calidad del proceso.
- **Processing layer:** Esta capa tendrá como misión principal las operaciones matemáticas necesarias tanto para extraer las características principales de un archivo de audio como para realizar el propio mastering con unos parámetros determinados.

3.2.1 Capa Front

A continuación procedemos a listar los requisitos necesarios para el desarrollo de esta capa:

- Diseño de una página web con formato *One-page*, con elementos visuales que denoten calidad y sean capaces de atraer al usuario, buscando la máxima profesionalidad en el diseño.
- División de la página en los submódulos de subir archivo, contacto, información y otros posibles elementos relacionados.
- Lógica para dar *click* a un botón y que nos permita seleccionar un archivo de audio.
- Lógica para enviar ese archivo de audio a la parte *Back-end*.
- Capacidad de recibir de la parte *Back-end* un archivo de audio y ponerlo a disposición del usuario para su descarga.

Como podemos observar, esta parte estará basada en el diseño del frontal de nuestra plataforma, por lo que priorizarán los elementos de diseño para la interacción con el usuario y la capacidad de enviar ese archivo para su procesado.

3.2.2 Capa Back

La capa *Back-end* es una de las más relevantes dentro de este desarrollo, ya que poseerá una parte del *core* de la aplicación y será la encargada de interconectar todas las capas. Por ello, analizaremos los requisitos de cada una de las conexiones con las diferentes capas para mostrar los requisitos individualizados por conexión.

Interconexión Back – Front:

- Capacidad de desplegar un servidor back que aloje nuestra página web.
- Ofrecer el routing necesario para acceder a la web.
- Recibir un archivo de audio de la parte frontal.
- Desarrollar la seguridad suficiente como para que no existan vulnerabilidades accesibles por el usuario.

Interconexión Back – Database

- Capacidad de conectarse y almacenar cualquier tipo de información dentro de una base de datos.
- Capacidad de leer de la base de datos y enviar la información obtenida a otros procesos.

Interconexión Back – Big data

- Capacidad de realizar peticiones a la capa *Big data*, esperar a los resultados obtenidos y redirigirlos hacia otros procesos o capas.

Interconexión Back – Processing

- Capacidad de enviar un archivo de audio en un formato determinado para la extracción de sus características.
- Capacidad de manejar la respuesta de la capa de procesado para orientarla a otras capas

Interconexión Back – IA

- Capacidad de nutrir de los datos necesarios al modelo de inteligencia artificial a desarrollar.
- Capacidad de conectarse con esta capa, pasarle un archivo de audio, esperar su procesado y devolver al cliente el archivo ya masterizado.

3.3.3 Capa DataBase

La capa que contiene la base de datos será la encargada de almacenar permanentemente la información registrada por todos y cada uno de los procesos de mastering que se realicen dentro de la plataforma. Para el análisis de la base de datos que necesitaremos, hay que tener en cuenta varios factores, como son el gran volumen de datos que vamos a manejar y el volumen de peticiones a las que se verá sometida.

Así, podemos sacar una serie de requisitos indispensables para nuestro desarrollo, como son:

- **Base de datos no relacional:** Al almacenar datos de diversa índole y que no están demasiado relacionados entre sí necesitaremos optimizar las consultas.
- **Escalabilidad vertical y horizontal:** A medida que nuestro proyecto crece, necesitamos tener la seguridad de que nuestra base de datos no es un impedimento para poder crecer con él.
- **Descentralización,** soporte de estructuras distribuidas.
- **Cambios de los esquemas en tiempo real.**
- **Posibilidad de ejecución en máquinas con pocos recursos**
- **Velocidad de escritura y lectura muy elevadas,** para aproximar el análisis posterior de toda la información al “real-time”.
- **Utilización de varios nodos,** necesitamos que nuestra información no solo sea persistente, si no que tenga redundancia para eliminar problemas con la seguridad y de back-ups.

3.3.4 Capa Big Data

La capa *Big Data* será la encargada de analizar todos los datos disponibles para obtener unos valores óptimos sobre los que aplicaremos todo nuestro procesado. Los principales requisitos serán los siguientes:

- Creación de algoritmos que permitan identificar los parámetros óptimos para las herramientas de mástering, basándose en los datos almacenados.
- Alta velocidad de procesado.
- Capacidad de visualizar los resultados de cada uno de los procesos en tiempo real.

3.3.5 Capa IA

A la hora de analizar esta capa, la hemos situado sobre la capa Back únicamente conectada a las bases de datos. Esto es así porque si bien es una más del conjunto, la idea con la que se ha diseñado es la de obtener un modelo de inteligencia artificial que nos permita sustituir todo el proceso de mastering anterior por algo mucho más sencillo y eficaz. Para ello, esta capa se nutrirá de un *dataset* etiquetado de archivos que tendremos en nuestra base de datos, mediante los cuales paralelamente a los procesos que se realicen dentro de la plataforma y una vez alcanzado una cantidad de datos suficiente, se creará el modelo mediante el entrenamiento de una red neuronal.

Una vez este modelo esté creado y funcione de forma óptima, reemplazará a la capa de *Big Data*, y será el encargado de proporcionar los valores de las variables necesarias para las herramientas de mastering. Así, conseguiremos eliminar el paso de tener que ejecutar los diversos algoritmos de la capa *Big data*, reduciendo el tiempo de espera del usuario y obteniendo una calidad similar.

Los requisitos fundamentales que tendrá esta capa por tanto serán los siguientes:

- Capacidad de entrenar un modelo de inteligencia artificial para el mastering de un archivo de audio a través de un conjunto etiquetado de procesos.
- Capacidad de realizar inferencias ante la entrada de un determinado archivo de audio.
- Capacidad de autoaprendizaje sobre los procesos exitosos o erróneos.

3.3.6 Capa Processing

La capa de procesado es la capa *core* de toda la aplicación, ya que será la encargada de extraer mediante diversos algoritmos y utilización de diversos módulos, las características principales de un archivo de audio, a la vez que también deberá ser capaz de modificar ese archivo en función de unos parámetros dados para realizar el mastering.

Las principales características que deberá tener esta capa son:

- Implementación de diversas herramientas de procesado de audio, como son ecualizadores, compresores y limitadores necesarios para realizar el proceso de masterización.

- Capacidad de recibir un archivo de audio, y mediante la utilización de esas herramientas obtener las características principales del archivo en un formato predefinido.
- Capacidad de modificar un archivo de audio en función de unos parámetros dados.
- Capacidad de funcionar como un micro servicio accesible a través de un *endpoint*.

CAPÍTULO 4: DISEÑO

En este capítulo se expondrá el diseño de la solución, como pasaba en el capítulo anterior nos encontramos ante un diseño complejo debido a la gran cantidad de tecnologías que se han elegido para llevar a cabo este proyecto. Por tanto, para la correcta exposición de los desarrollos procederemos a explicar una vista general de toda la solución y posteriormente el detalle de cada uno de los módulos o micro servicios utilizados.

4.1 Arquitectura general de la solución

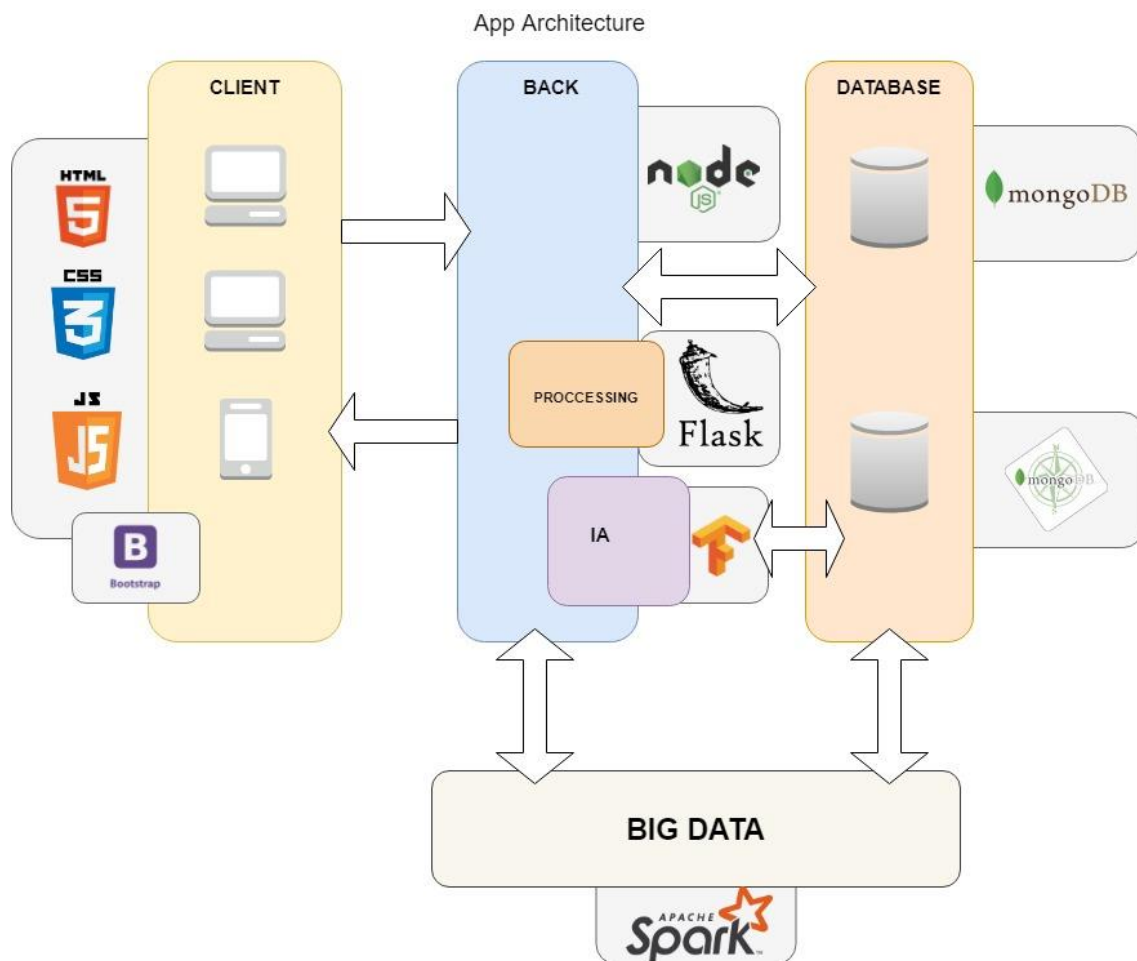


Figura 10. Arquitectura general de la solución.

Tras una compleja investigación, hemos conseguido obtener las tecnologías necesarias que satisfacen todos los requisitos que se nos plantearon en el capítulo de análisis. Como podemos observar en la Figura 10, hemos elegido un diseño modular de nuestra plataforma web, de tal manera que cualquiera de los módulos podría ser sustituido por otro similar sin que afectase al resto, creando una aplicación más robusta y escalable.

Siguiendo este esquema, el proceso general que seguiría un usuario desde que entra en la plataforma web hasta que recibe el archivo masterizado sería el siguiente:

- El usuario entra en la plataforma web a través de una *url*, el servidor le manda una página web navegable con la que interactuar.
- El usuario sube un archivo de música al servidor.
- El servidor manda el archivo al micro servicio de procesado.
- Se extraen las características principales del archivo y se mandan de vuelta al servidor.
- Se almacenan esas características iniciales y se mandan al módulo *Big Data*
- Se procesan los datos con los algoritmos creados para extraer los mejores parámetros que aplicar a las herramientas de procesado para la correcta masterización del archivo.
- Se devuelven los parámetros al servidor.
- Se almacenan los parámetros óptimos, y se envían al micro servicio de procesado.
- Se masteriza el archivo con esos parámetros.
- Se devuelve el archivo al módulo back.
- Se devuelve el archivo al usuario.
- Se recoge el feedback del usuario.
- Se almacena el feedback conjuntamente a los datos de todo el proceso.

4.2 Diseño Front

El módulo frontal será el encargado de interactuar directamente con el usuario, por ello hay dos factores que serán clave a la hora de diseñarlo, **la usabilidad y la sencillez**.

Así, crearemos una página web sencilla y amigable que constará de 5 segmentos, divisiones de la propia página web que nos mostrarán distinta información recolectada

por la plataforma así como métodos de contacto, intentando asemejar la calidad del resultado final a una página web profesional.

4.2.1 Segmento de presentación



Figura 11. Segmento “conócenos”

Dentro de este segmento el usuario podrá encontrar información relacionada con la plataforma web, qué es lo que hacemos y cómo lo hacemos, nuestra experiencia en el sector y la gratuidad del servicio. Este segmento nos servirá para dar veracidad a nuestra página web exponiendo nuestros factores diferenciales.

4.2.2 Segmento funcionamiento



Figura 12. Segmento “como funciona”

Explicación simplista del proceso que el usuario ha de seguir para poder utilizar la aplicación, mostramos también estadísticas de los géneros musicales que más se han masterizado en nuestra plataforma.

4.2.3 Segmento estadísticas



Figura 13. Segmento “contador”

En este segmento mostramos una serie de estadísticas, como son:

- El número de *tracks* masterizados en la plataforma
- El número de artistas analizados y cuyas características aparecen en nuestra base de datos. (Y de las cuales nos nutrimos para realizar nuestros procesados).
- El número de personas que ha visitado la página
- El coste por *track*, al ser un servicio gratuito queremos hacer énfasis en que será “0”.

4.2.4 Segmento de contacto



The image shows a contact form titled 'CONTÁCTANOS' with the subtitle 'Si tienes alguna duda o petición ¡siéntete libre de contactarnos!'. It includes contact details for 'Oficina' (Universidad Carlos III de Madrid, +34 632 154 555, info@onmastering.com) and 'Horario laboral' (laborables 9:00 to 16:00, Fin de semana Cerrado). There is a form with fields for 'Nombre', 'Email', and 'Mensaje', and an 'Enviar' button.

Figura 14. Segmento “contáctanos”

Segmento típico de una página web en el que el usuario podrá obtener los datos del desarrollador y cómo ponerse en contacto con él.

4.2.5 Segmento subir Archivo



Figura 15. Vista inicial de la web “subir archivo”

Es el segmento principal de la página, se ha optado por mantener en primer plano el botón de subir archivo, ya que es el foco principal de toda nuestra aplicación.

4.2.6 Lógica de la página

Si bien nos encontramos ante un formato *One-page*, sin necesidad de añadir un *routing* específico para más de una página, podemos encontrar una mínima lógica diseñada para facilitar la interacción del usuario por la página.

- **Navegación por secciones:** En la esquina superior derecha, tendremos acceso mediante un *click* a las diferentes secciones de la página, evitando que el usuario pierda tiempo desplazándose hacia ellas.
- **Botón flotante:** A medida que descendemos en la página, nos aparece un botón flotante que nos permite volver al inicio de la página.
- **Estadísticas:** Cada una de las estadísticas reflejadas en dicha sección es un valor tomado de nuestra base de datos y reflejado en nuestro frontal, mediante una serie de *Querys* o peticiones, logramos conocer los datos antes de cargar la página para posteriormente mostrarlos al usuario.
- **Subir Archivo:** Este botón nos permite abrir una ventana de diálogo en nuestro navegador para explorar la carpeta donde tenemos el archivo de audio que queremos procesar, facilitándonos el proceso de seleccionar el archivo. También se encarga de una vez realizada la subida, transformar el formato del archivo a *base64* y enviarlo hacia el módulo back-end.

4.2.7 Tecnologías utilizadas

Para el diseño de este frontal, se ha investigado sobre las últimas tecnologías en diseño web, las cuales pasamos a nombrar a continuación (ver Figura 16).



Figura 16. : Zoom al módulo front

- **Html5:** Última versión de la herramienta de *html*, nos ha ayudado a la hora de estructurar la página web.
- **CSS:** Las Hojas de estilo en cascada nos permiten la separación del contenido del documento con respecto de la presentación del documento.
- **Javascript:** Lo utilizaremos para desarrollar toda la lógica planteada en el punto 4.2.6.
- **Bootstrap:** *Framework* de *CSS* y *html5* consistente en un conjunto de herramientas de código abierto en la que podemos encontrar plantillas con un diseño profesional. Es utilizado actualmente por la *NASA* para el desarrollo de sus frontales [12].

4.3 Diseño Back

Como se planteó en el capítulo de análisis, el módulo back de nuestra aplicación será el encargado de interconectar todos y cada uno de los módulos, a la vez que sirve nuestro frontal.

Para determinar cuál es la tecnología que nos permitirá conseguir los mejores resultados, primero debemos realizar una tarea de investigación sobre los lenguajes de programación más utilizados a la hora de desarrollar nuestro *Back-end*, y qué mejor manera de comprobar este dato que acudiendo a una de las páginas más importantes para los desarrolladores a nivel global, como es *stackoverflow*:

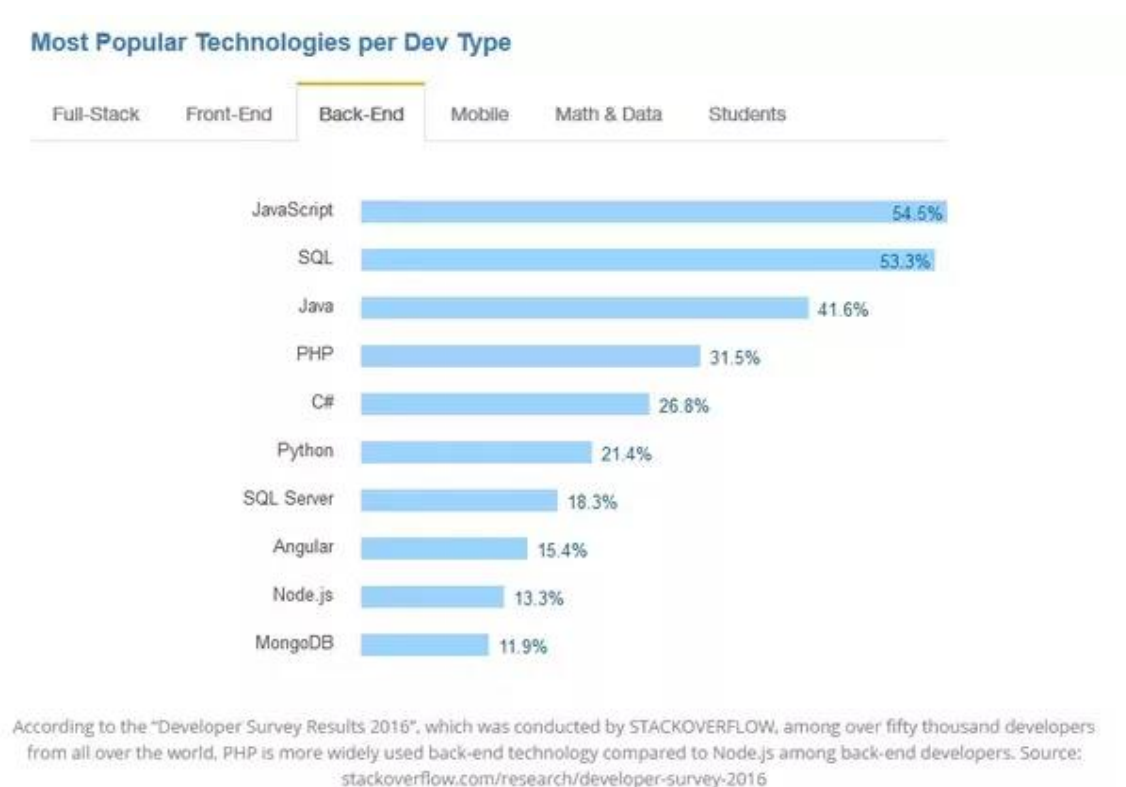


Figura 17. Lenguajes de programación Back-End más populares [13]

Como podemos observar en la Figura 17, javascript es uno de los lenguajes de programación más utilizados, por lo que se utilizará para el desarrollo de nuestro módulo back-end. Optaremos en nuestro caso por un *Framework* específico llamado *NodeJS*, el cual atendiendo a su propio reporte es uno de los más utilizados para el desarrollo de *Back-ends* (ver Figura 18).

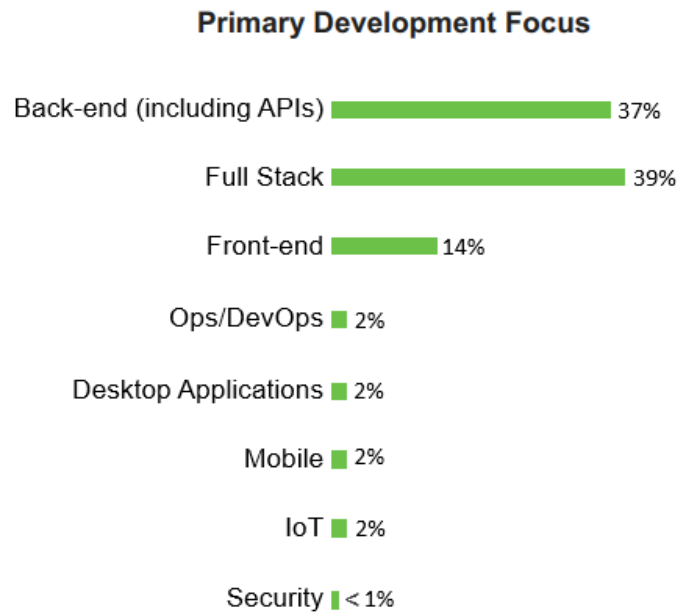


Figura 18. Focos de los desarrollos en NodeJS [14]

Por tanto, atendiendo a la página web oficial del *framework*, podemos comprobar que posee una serie de requisitos por los que realizaremos esta elección, como son:

- Capacidad de crear aplicaciones de forma rápida y robusta.
- Velocidad de procesamiento de peticiones superior a sus competidores.
- Orientado a *streams* de datos.
- Destinado a la creación de aplicaciones web que funcionan “en tiempo real”.
- Facilidad de conexión con bases de datos no relacionales.
- Gestor de paquetes potente, con una comunidad enorme tras él.
- Orientado a la creación de APIs.
- Facilidad para ser alojado en servidores como los de *Amazon Web Services*.

Todas estas características son necesarias para crear un módulo que cumpla los requisitos que nos planteamos en el análisis de la capa, obteniendo así 3 partes diferenciadas dentro del diseño del mismo: sistema de rutas de la aplicación, conexión con la base de datos y conexión con el resto de micro servicios (ver Figura 19).

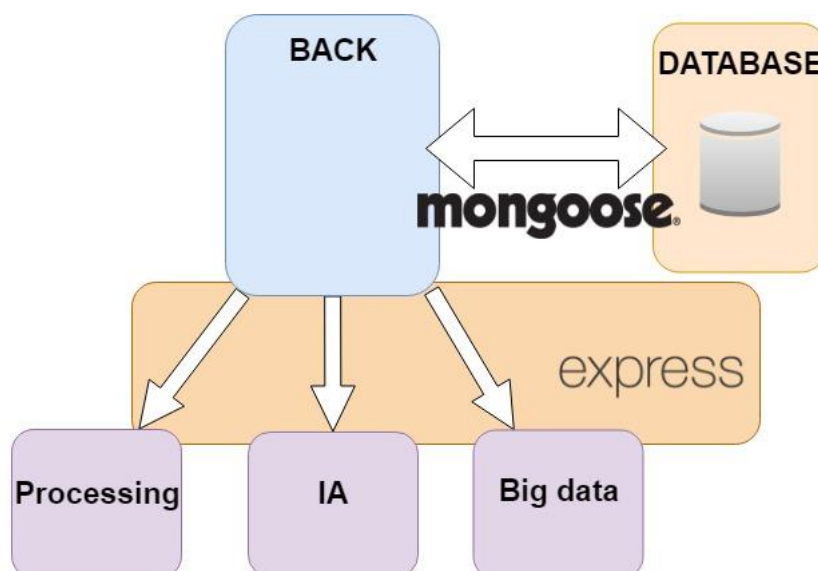


Figura 19. Arquitectura Back-end en detalle

4.3.1 Routing

El *routing* en nuestra aplicación consistirá en nombrar una serie de *end-points* que permitirán enlazarnos con el resto de módulos. La conexión con estas direcciones se realizará a través de mensajes POST, exceptuando la obtención del frontal que funcionará a través de un mensaje tipo GET. Para el desarrollo de este routing, nos valdremos de un *framework* específico de *NodeJS* llamado *Express* (Figura 18), el cual nos permitirá crear todo de una forma muy sencilla, otorgándonos la capacidad de parsear los mensajes a través de la instalación de diversos módulos como el *bodyparser*. A continuación pasaremos a definir todas las direcciones que contendrá nuestro back-end, su funcionalidad, y el modelo de mensaje que será capaz de procesar [15].

❖ “localhost:5555/”

Será nuestra dirección local, a través de la cual obtendremos nuestra web. Accederemos a ella a través de nuestro navegador, el cual recibirá una petición web cuya respuesta será nuestro frontal. El puerto elegido para que nuestro servidor se quede escuchando es el 5555, pero será parametrizable.

❖ “/process”

Bajo esta dirección, podremos iniciar todo el proceso de mastering del audio. En esta ruta recibiremos un mensaje tipo POST, cuyo cuerpo deberá ir encapsulado en un formato JSON con las siguientes características:

```
{  
  "data": "9/Jk=..."  
}
```

Figura 20. Modelo de datos JSON de la capa Back-end

Recibiremos por tanto un mensaje en formato clave/valor en el que encontraremos los datos de nuestro archivo de audio codificados en formato *base64* (ver Figura 20) El motivo de la elección de esta codificación es debido a que a la hora de realizar peticiones POST con un cuerpo en formato JSON, la inclusión de un *array de bytes* en el mensaje debe estar codificada para su correcto funcionamiento.

Una vez recibida la petición, nuestro back funcionará como una especie de mensajero, y dirigirá el mensaje en el mismo formato que lo recibió hacia el módulo de procesado, cuyo *endpoint* podremos encontrar en el apartado 4.5 de esta memoria, a través de una petición POST.

Una vez recibida la respuesta, se proseguirá con la cadena de masterización, conectando con el resto de módulos para obtener un archivo final masterizado. Así, la respuesta final que obtendremos de esta ruta será un archivo final descargable por el usuario.

4.3.2 Conexión con la base de datos

En este apartado haremos una pequeña mención especial para hablar de la tecnología que utilizaremos como conector, *Mongoose*.

Mongoose es un módulo de *NodeJS* que nos permite modelar objetos que posteriormente serán almacenados en una base de datos no relacional, en este caso *MongoDB*, permitiéndonos:

- Creación rápida y sencilla de modelos de datos.
- Castings automáticos durante la creación del modelo.

- Validaciones.
- Construcciones sencillas de *querys*.

Así, podremos almacenar cualquier tipo de dato de una forma sencilla y con mayor rapidez que con otro tipo de tecnologías [16].

4.4 Diseño Database

Nuestra base de datos será el punto donde almacenaremos todos los datos recogidos a lo largo de los diferentes procesos de masterización.

Tras el análisis realizado en el capítulo 3.2.3 (requisitos necesarios) llegamos a la conclusión de que podemos encontrar una solución que cumpla esos requisitos a través de la base de datos no relacional *MongoDB* [17]. Las características principales de esta tecnología las podemos extraer de su propia página web, como son:

- Soporte para *querys* ad-hoc.
- Indexado.
- Duplicado de datos.
- Balance de carga.
- Soporta métodos *map reduce*, necesarios para el *Big data*.
- Base de datos sin esquema predeterminado.
- Formato de datos *BSON*.
- Velocidad de lectura/escritura elevada.

Así, en esta parte tendremos 2 grandes bloques:

- Arquitectura de la base de datos: Número de bases de datos, colecciones y función de cada una de ellas.
- Modelo de datos de cada una de las colecciones: Tipos, características, formatos, etc.

4.4.1 Arquitectura de la base de datos

Para este proyecto, tendremos 2 bases de datos principales, las cuales serán copias idénticas para añadir seguridad a través de un factor de redundancia. Solo se escribirá y leerá de una de ellas, permaneciendo su contraria como un back-up sobre el que se volcarán todos los datos periódicamente.

La arquitectura de la base de datos se muestra a continuación:

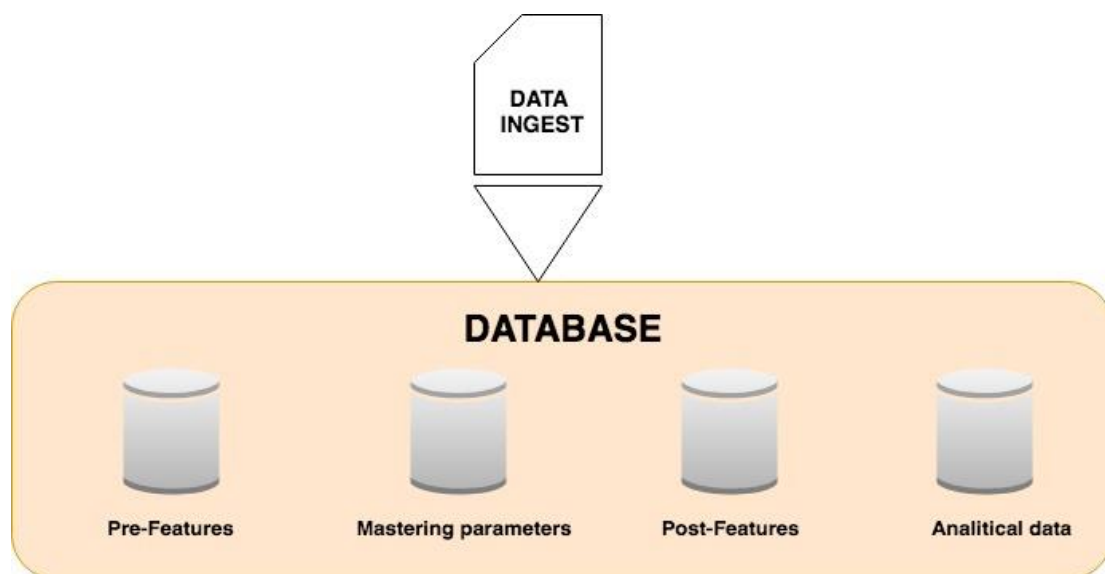


Figura 21. Arquitectura de la base de datos

Como podemos observar en la Figura 21, tendremos una base de datos compuesta de 4 colecciones (las colecciones en *MongoDB* serán la equivalencia directa con las tablas en las bases de datos relacionales), las cuales pasamos a definir a continuación:

- **Pre-features:** Colección donde se almacenarán las características principales de un archivo de audio antes de su masterización.
- **Mastering parámetros:** Colección donde se almacenarán los resultados obtenidos del módulo de big data en cada iteración de la cadena de mastering.
- **Post-features:** Colección donde se almacenarán las características principales de un archivo de audio una vez ha pasado a través de todo el proceso de mástering.
- **Analitical data:** Colección donde se almacenarán datos como número de procesos, *timestamps*, éxitos o fracasos, etc.

4.4.2 Modelo de datos de las colecciones

En este apartado pasaremos a describir los diferentes modelos de cada una de las colecciones que hemos definido anteriormente. Toda la información recogida se almacenará en un formato JSON clave/valor, por lo que iremos definiendo uno a uno los campos necesarios que lo conformarán para cada una de las colecciones.

4.4.2.1 Pre-features & Post-features

```
{
  "rms": -28,
  "rate": 44,
  "mean_db": {
    "band_1" : -28,
    "band_2" : -24,
    "band_3" : -34,
    "band_4" : -28,
    "band_5" : -50,
    "band_6" : -27,
    "band_7" : -23,
    "band_8" : -15,
    "band_9" : -34,
    "band_10" : -40
  },
  "highest_peak" : -3
}
```

Figura 22. Modelo de datos de la colección Pre/Post-features

Como podemos observar en la Figura 22, tenemos cuatro valores que nos proporcionarán las características principales de nuestro tema musical, como son:

- **RMS:** valor promedio medido en decibelios de nuestro archivo musical.
- **Rate:** frecuencia de muestreo en kHz.
- **Mean_db:** valor promedio en decibelios de cada una de las 10 bandas definidas.
- **Highest_peak:** valor en decibelios del pico más alto de sonido en nuestro archivo.

4.4.2.2 Mastering parameters

```
{
  "id" : 1545,
  "rms": -28,
  "rate": 44,
  "eq":
  {
    "band_1_gain" : -28,
    "band_2_gain" : -24,
    "band_3_gain" : -34,
    "band_4_gain" : -28,
    "band_5_gain" : -50,
    "band_6_gain" : -27,
    "band_7_gain" : -23,
    "band_8_gain" : -24,
    "band_9_gain" : -34,
    "band_10_gain" : -40
  },
  "compressor":
  {
    "threshold" : 0.6,
    "delay" : 40,
    "signal_length" : 1,
    "release_coeff" : 0.9995,
    "attack_coeff" : 0.9
  },
  "limiter":
  {
    "threshold" : 0.6,
    "delay" : 40,
    "signal_length" : 1,
    "release_coeff" : 0.9995,
    "attack_coeff" : 0.9
  }
}
```

Figura 23. Modelo de datos de la colección Mastering Parameters

En esta colección tendremos en cuenta todo el proceso de la cadena de mastering almacenando los valores para cada uno de los elementos de la cadena, así podemos encontrar (ver Figura 23):

- **Id** : identificador del proceso.

- **RMS**: valor promedio final objetivo, aquel que queremos conseguir para nuestro archivo de salida mediante los parámetros incluidos en el cuerpo del JSON.
- **Rate**: frecuencia de muestreo.
- **Eq**: parámetros relacionados con la ganancia a aplicar en el ecualizador paramétrico.
- **Compressor**: parámetros a aplicar al compresor diseñado en la cadena.
- **Limiter**: parámetros a aplicar al limitador diseñado en la cadena, si bien cabe mencionar que son similares a los del compresor debido a que la limitación es un tipo de compresión.

4.4.2.3 Analitical Data

```
{
  "id" : 1545,
  "timestamp_init": "2018-09-13T17:00:00+00:00",
  "timestamp_end": "2018-09-13T17:00:00+00:00",
  "time_delayed" : 25,
  "success" : true
}
```

Figura 24. Modelo de datos de la colección Analitical Data

Como hemos indicado anteriormente, en esta colección almacenaremos los datos más relevantes con respecto de cada uno de los procesos (ver Figura 24).

- **Id** : identificador del proceso
- **Timestamp_init**: Momento temporal en formato ISO 8601 en el que comenzó el proceso.
- **Timestamp_end**: Momento temporal en el que finalizó el proceso.
- **Time_delayed**: Tiempo en segundos que ha tardado en completarse todo el proceso.
- **Success**: *Flag* que indicará cuando un mastering haya sido exitoso recogiendo el feedback del usuario.

4.5 Diseño Procesado

El módulo de procesado de audio es probablemente el más complejo a nivel técnico de toda nuestra plataforma. Si bien hemos hablado hasta ahora de conectores, módulos de almacenamiento y otro tipos de procesos, es aquí donde se desarrollará toda la cadena de mastering, por lo que tendremos que nutrir de las herramientas necesarias a este micro servicio para que sea capaz de satisfacer todos los requerimientos técnicos.

Como hemos comentado anteriormente, el módulo de procesado constará de un micro servicio accesible vía REST API, ya que en nuestro caso queremos conseguir que cada una de las herramientas sea accesible por separado, eliminando la necesidad de tener que utilizar todos los elementos de la cadena en el proceso y pudiendo obviar alguno de ellos si lo considerásemos oportuno.

En este apartado hablaremos de la tecnología que utilizaremos en el diseño del micro servicio, su arquitectura y el diseño de cada una de las herramientas necesarias para su desarrollo.

4.5.1 Micro servicio de mastering

Uno de los mayores problemas que encontramos a la hora de realizar este proyecto, fue encontrar un lenguaje de programación que tuviese la suficiente potencia para poder procesar archivos de audio a la vez que exponer una serie de servicios. La solución la encontramos en la herramienta *Flask*, un *framework* de *Python* mediante el cual podremos realizar un routing hacia cada una de las herramientas, creando un micro servicio modular con elementos independientes pero fácilmente accesibles [18].

Pasamos así a mostrar un plano de la arquitectura general del micro servicio, para posteriormente analizar y explicar su composición:

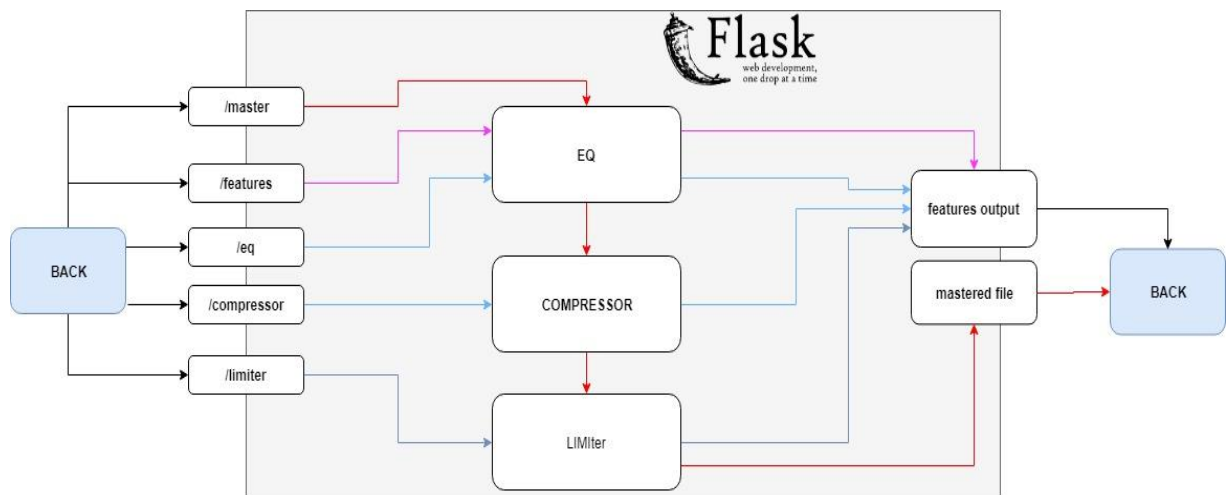


Figura 25. Arquitectura del micro servicio de mastering

En la Figura 25 podemos observar dos partes muy diferenciadas dentro de la arquitectura, como son el routing y el conjunto de herramientas.

4.5.2 Routing

Gracias a *Flask* podremos crear un sistema de routing que constará de cinco rutas diferentes:

- **“/features”:**

Será la ruta a través de la cual extraeremos las características principales de nuestro archivo, estará conectada directamente con el ecualizador ya que se valdrá de éste para extraer esas características.

- **“/eq”:**

Conectará con el ecualizador, en base a unos parámetros nos devolverá un tema ecualizado.

- **“/compressor”:**

Será la conexión con el compresor, recibirá unos parámetros de entrada que nos permitirán recibir un archivo comprimido.

- **“/limiter”**

Conexión con el limitador, al igual que con las herramientas anteriores, recibirá una serie de parámetros de entrada que nos permitirán limitar el archivo de audio.

- “/master”

La ruta master, será la encargada de recibir un archivo y consecuentemente ir enviándolo a los distintos módulos de procesado, devolviendo el archivo de audio final masterizado.

4.5.3 Herramientas de mastering

A continuación pasamos a detallar cada una de las herramientas que vamos a utilizar en nuestro proceso de mastering:

4.5.3.1 Ecualizador paramétrico

Durante todo el desarrollo hemos hablado de las características más importantes que definen a un tema y que nos servirán para poder concluir que parámetros serán los correctos para poder masterizarlo. Así, extraeremos cuatro elementos de nuestro archivo de audio con los que trabajaremos posteriormente: El RMS total, el valor de pico, la frecuencia de muestreo y la media de las potencias en las diferentes bandas.

Para garantizar que tenemos el máximo detalle de cada una de las bandas de nuestro archivo de audio dividiremos nuestro espectro en 10 bandas, teniendo en cuenta que las frecuencias audibles con las que trabajaremos se encuentran en el rango entre 20Hz-20Khz, y dividiendo el espectro por octavas, obtendremos:

```
{  
  "band_1" : 63,  
  "band_2" : 125,  
  "band_3" : 250,  
  "band_4" : 500,  
  "band_5" : 1000,  
  "band_6" : 2000,  
  "band_7" : 4000,  
  "band_8" : 8000,  
  "band_9" : 16000,  
  "band_10" : 20000  
}
```

Figura 26. División por bandas del espectro audible

Los valores de las bandas que encontramos en la Figura 26 están expresados en Hz, con esta herramienta seremos capaces de extraer el valor promedio en dB de cada una de las bandas para posteriormente crear un objeto que devolveremos como resultado final.

Para obtener estos resultados, crearemos en *Python* una serie de filtros de pico que abarquen cada una de las frecuencias que hemos definido (ver Figura 27).

```
##### Difference equation for Peak filter #####
self.fc_p = 20000
self.V_p = 1 #gain in dB
self.Q = 10

K_p = math.tan(math.pi * self.fc_p / self.RATE)
G_p = 10**(self.V_p/20)

self.b6 = (1 + (G_p/self.Q)*K_p + K_p**2)/(1 + (1/self.Q)*K_p + K_p**2)
self.b7 = (2*((K_p**2) - 1))/(1 + (1/self.Q)*K_p + K_p**2)
self.b8 = (1 - (G_p/self.Q)*K_p + K_p**2)/(1 + (1/self.Q)*K_p + K_p**2)
self.a5 = (2*((K_p**2) - 1))/(1 + (1/self.Q)*K_p + K_p**2)
self.a6 = (1 - (1/self.Q)*K_p + K_p**2)/(1 + (1/self.Q)*K_p + K_p**2)
```

Figura 27. Ejemplo de filtro de pico en Python centrado en 20Khz

Así, aplicando estos filtros con una ganancia 1 (no tendrán efecto sobre la ganancia de la banda) conseguiremos obtener los dBs correspondientes a cada rango de frecuencias, y por tanto ser capaces de extraer las características principales que buscábamos.

Si la acción que queremos realizar es la de mástering en lugar de la de extracción de características, el proceso será algo diferente, teniendo que aplicar las ganancias por banda que recibimos como parámetro al archivo de audio, consiguiendo así la ecualización deseada.

Cabe destacar en este módulo que las características extraídas pueden no ser suficientes en nuestro caso para determinar con total exactitud cómo es un tema musical, pero serán suficientes para mostrar el planteamiento y como base para seguir incluyendo parámetros.

4.5.3.2 Compresor

El compresor será la pieza que actúe sobre el rango dinámico de nuestro archivo de audio, vendrá definido por 5 características principales:

- **Threshold:** Valor en dB a partir del cual nuestro compresor empezará a actuar

- **Attack_coeff:** Tiempo que tardará nuestro compresor en actuar.
- **Release_coeff:** Tiempo que tardará nuestro compresor en dejar de actuar.
- **Delay:** Variable en segundos que nos dirá cuanto tiempo queremos posponer la compresión
- **Ratio:** Variable que nos dice cuánto se comprimirá la señal.

Existen ciertos parámetros del compresor que en una primera iteración serán asignados por defecto y cuyo valor no será parametrizable, como es el caso de la variable *knee*.

4.5.3.3 Limitador

La limitación es una forma de compresión por lo que los valores que encontraremos dentro de esta sección serán los mismos que los del compresor. Los valores de ataque serán más cercanos a 0 ya que buscaremos que empiece a actuar cuanto antes, los valores de ratio serán también elevados, ya que nuestro objetivo final es acercar el valor RMS promedio a 0 sin distorsionar la señal, como podemos observar en la Figura 28.

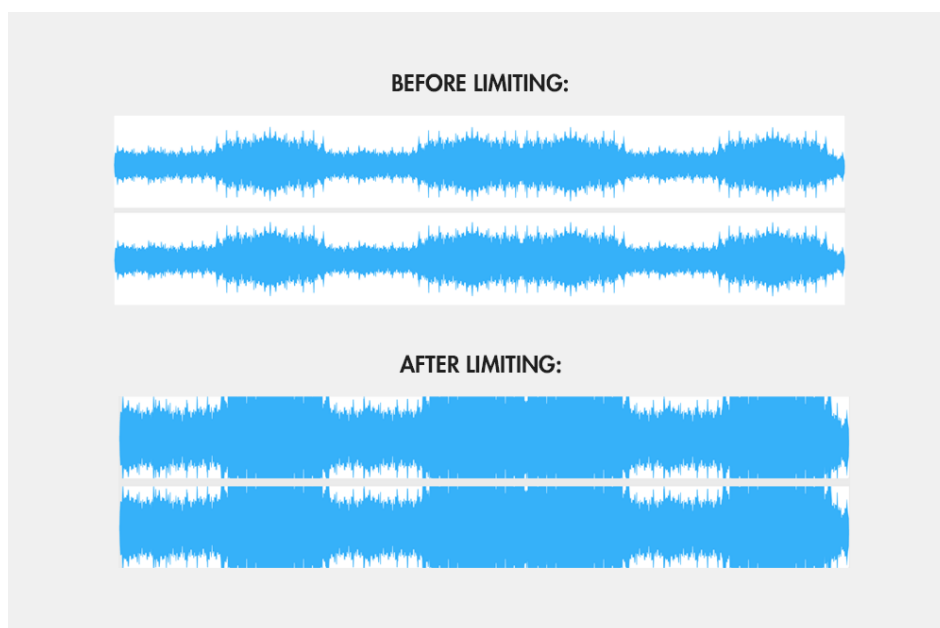


Figura 28. Audio antes y después de la limitación

Si bien la limitación es uno de los procesos más sencillos, es a la vez uno de los que más efecto tiene a la hora de medir la calidad de un master debido a la psicoacústica de

la percepción del volumen, haciendo que un tema que se escucha más alto pueda aparentar poseer una calidad mayor.

4.6 Diseño Big Data

Dentro de nuestro módulo de Big data, la misión principal será la de manejar la gran cantidad de información que tenemos almacenada en nuestra base de datos. Para ello, dividiremos este módulo en dos partes: Lógica de tratamiento de los datos y visualización *Real-time* de los mismos.

Para conseguir satisfacer los requisitos que proponíamos en el análisis, usaremos una de las mayores plataformas actuales para el procesamiento de datos, *Spark* [19]. La cual posee características como son:

- **Velocidad:** mediante la explotación de optimizaciones en memoria, *Spark* es capaz de ser hasta 100 veces más rápido que la ejecución del algoritmo *MapReduce* en *Hadoop*.
- **Framework unificado:** Incluye librerías de alto nivel, como soporte para SQL, *machine learning*, procesamiento de *streams* y gráficos, etc.
- **Simplicidad:** Incluye una serie de APIs fáciles de usar para operar con grandes conjuntos de datos, que contiene una serie de operadores muy sofisticados para transformar y manipular datos semi-estructurados.
- **Conectividad con MongoDB:** Incluye también un conector específico para la base de datos que utilizaremos en nuestro proyecto.

Como podemos observar, este *framework* está destinado al tratamiento de los datos *Real-time* debido a su gran velocidad de procesamiento, lo que nos hará seleccionarlo por encima de su principal competidor en este segmento (*Hadoop*), debido a que uno de los objetivos de nuestra plataforma es la velocidad a la que entregaremos un resultado.

4.6.1 Lógica de tratamiento de los datos

Para procesar todos los datos, diseñaremos un algoritmo principal que se encargará de proporcionarnos los valores óptimos para el mastering de nuestro archivo de audio: “Real Time Moving Average Clustering” (RTMAC) [20].

Si bien nuestra lógica nos diría que una de las formas más sencillas de solucionar este problema sería mediante el método *K-means*, cuando hablamos de real time nos encontramos con un problema, y es que la inclusión de nuevos datos que siempre están llegando hacen que el comportamiento del proceso pueda cambiar con el tiempo. Así, el parámetro “k” es desconocido, y puede cambiar para diferentes aplicaciones. También, nos encontraremos que no existe un mecanismo para olvidar los datos antiguos, a medida que se van creando registros cuyos parámetros se encuentran mejor optimizados, los registros antiguos empiezan a ser un problema porque sus pesos no están bien distribuidos.

Otro de los problemas que nos encontramos es que cada nuevo proceso que realizamos tiene un número indeterminado de iteraciones, lo que supondrá un problema ya que no podremos conocer el tiempo aproximado que tardará nuestro algoritmo en encontrar la solución, alejándonos de nuestro objetivo de conseguir un resultado lo más aproximado al tiempo real.

Por tanto la solución es usar RTMAC, una variante de *K-means* que nos permite para cada caso:

1. Encontrar la agrupación con el centroide más próximo.
2. Incluir el nuevo proceso en la agrupación.
3. Si esa agrupación tiene actualmente el tamaño máximo de nMax(máximo número de procesos por agrupación), elimina el proceso más antiguo de la agrupación.
4. Vuelve a calcular los nuevos valores de los centroides.

Una vez que hemos clasificado correctamente el punto en el que se encuentra nuestro archivo según sus características, tendremos que centrarnos en sus vecinos cercanos y conocer qué parámetros han utilizado para obtener un mástering exitoso, consiguiendo así encontrar una relación entre los parámetros de entrada y los parámetros necesarios para la correcta masterización del archivo de audio en función de los procesos previos.

Sin embargo, surge un gran problema a priori, y es que el número de variables es un factor fundamental para el correcto funcionamiento del algoritmo. Categorizar procesos con 2 variables es algo sencillo e incluso a simple vista podríamos hacerlo en un mapa de 2 dimensiones, pero cuando hablamos de 10 variables la cosa se complica, ya que

calcular todo este proceso para 10 dimensiones puede ser extremadamente complejo. De ahí que la idea final sea la de sustituir este módulo por un bloque de inteligencia artificial para agilizar todos los procesos.

4.6.2 Visualización y analítics

Una de las características principales y diferenciales de nuestra aplicación será la posibilidad de observar los procesos en tiempo real, pudiendo acceder a todos los datos que se han generado en cada uno de nuestros procesos a través de una interfaz gráfica. Al diseñar nuestra base de datos orientada a objetos con un id único por proceso, podremos relacionar fácilmente cada uno de los campos almacenados sin la necesidad de realizar *queries* específicas para extraer la información.

Para conseguir esto recurriremos a *Compass*, la herramienta de la suite de MongoDB que nos permitirá acceder a todo el contenido [21], permitiéndonos:

- Crear gráficos en tiempo real sobre cada uno de los procesos de mastering.
- Visualizar los procesos de mastering erróneos.
- Visualizar los resultados de nuestro algoritmo.
- Crear gráficas sobre el feedback recibido por los usuarios.
- Controlar el estado del servidor y su correcto funcionamiento.
- Calcular el porcentaje de mastering realizados con éxito.
- Elaborar un informe automático con las gráficas más relevantes.

Gracias a esta herramienta podremos observar en qué partes nuestra aplicación es menos robusta y qué nos está generando problemas para elaborar una estrategia de acción sobre los algoritmos que se han creado. Así, conseguiremos tener un control absoluto sobre lo que está pasando y seremos capaces de extraer conclusiones rápidas para tomar decisiones eficaces (ver Figura 29).

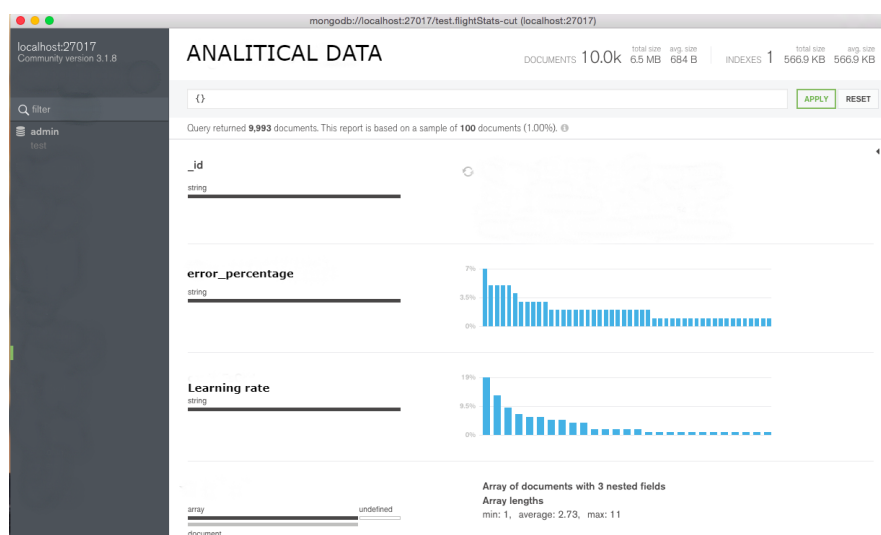


Figura 29. Ejemplo de visualización de datos en Compass

4.7 Diseño IA

Nuestro módulo de inteligencia artificial será un elemento externo a nuestro proyecto cuya finalidad será la de sustituir a toda la capa de Big data, a través del entrenamiento de un modelo que nos permita inferir los parámetros para la realización del mástering automáticamente, sin la necesidad de realizar una cantidad elevada de operaciones y adaptándose a las características de cualquier archivo a procesar.

Uno de los mayores problemas identificados a la hora de calcular los parámetros con el algoritmo mencionado en el capítulo 4.6.1, es que en ocasiones el número de variables que debe procesar para obtener un resultado es tan elevado, que puede hacer que los tiempos que manejemos se alejen demasiado del planteamiento original de nuestra solución, creando una experiencia de usuario insatisfactoria independientemente del resultado final del mastering.

Por ello decidimos que una vez se procesasen suficientes elementos como para obtener un conjunto de datos que nos permitiese realizar este entrenamiento, se llevaría a cabo para obtener un modelo mucho más rápido que realizase el mismo trabajo.

Existen tres grandes bloques en el diseño de este módulo, en los que se desarrollará tanto la forma en la que obtendremos nuestros datos, como las variables que hemos tenido en cuenta a la hora de crear el modelo y su implantación para inferencias.

La tecnología utilizada para el desarrollo de este módulo será *Tensorflow*, debido a que es una plataforma destinada al desarrollo de inteligencia artificial que nos proporciona una API fácil de usar y con una potencia suficiente como para cumplir los requisitos que planteamos en el análisis.

4.7.1 Obtención del dataset

Una de las partes más importantes del desarrollo de un modelo de inteligencia artificial es la obtención de unos datos que nos permitan obtener un resultado veraz, para ello las características que tienen que tener estos datos en nuestro caso particular son:

- Datos correctamente estructurados para su procesado.
- Datos etiquetados
- Separación de los datos en dos carpetas, “/train” y “/test”

Los datos almacenados en nuestra carpeta “/train” serán aquellos que utilizaremos para entrenar nuestro modelo, mientras que los almacenados en la carpeta “/test” serán los datos con los que comprobaremos que la eficacia con la que nuestro modelo funciona será superior a un umbral que ajustaremos nosotros mismos en función de los requisitos de nuestra aplicación.

Los datos que utilicemos para el entrenamiento y el testeo deberán estar etiquetados, es decir, adjuntaremos un archivo que nos dirá para cada elemento del conjunto de datos si el proceso de mástering ha sido exitoso o no y los parámetros necesarios para lograr replicar el proceso, para que posteriormente nuestro modelo pueda ser testado y evaluado.

La estructura de los datos, será equivalente a la mostrada en la Figura 23, por lo que utilizaremos como valores de entrada las características iniciales de un proceso, y como valores de salida aquellos parámetros necesarios para la correcta masterización del archivo de audio.

4.7.2 Entrenamiento del modelo

Para el entrenamiento de nuestro modelo utilizaremos una red neuronal artificial (ver Figura 30), la cual será entrenada mediante el algoritmo *Back propagation*.

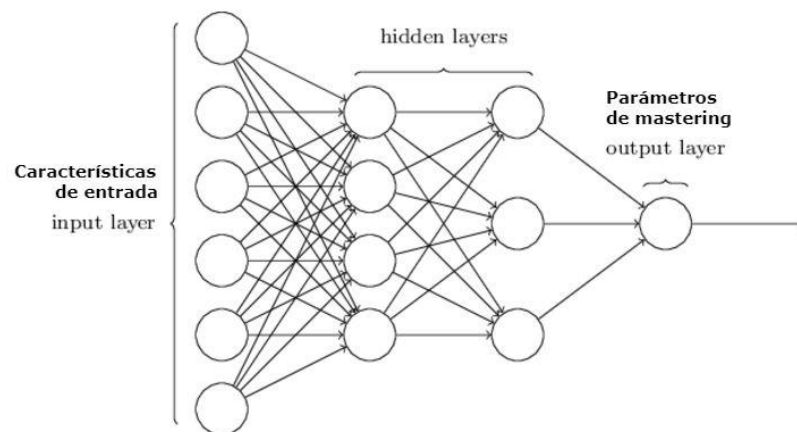


Figura 30. Ejemplo de red neuronal

Así, los parámetros de entrada de nuestra red (*input layer*) serán las características extraídas de un archivo musical, la activación de las neuronas nos proporcionará un resultado final que contendrá los parámetros necesarios para el mastering (*output layer*).

Si bien es cierto que el algoritmo *Back propagation* conlleva un tiempo de entrenamiento elevado, no será un factor relevante debido a que este proceso será realizado en paralelo, por lo que los dos únicos factores relevantes y que deberemos tener en cuenta serán el tiempo de inferencia y el porcentaje de éxito.

4.7.3 Inferencia

La inferencia es el proceso mediante el cual a través de unos parámetros de entrada, nuestro modelo obtendrá unos parámetros de salida.

Una vez nuestro modelo ha sido entrenado, procederemos a sustituir el Módulo de Big Data, obteniendo una arquitectura final como podemos ver en la Figura 31:

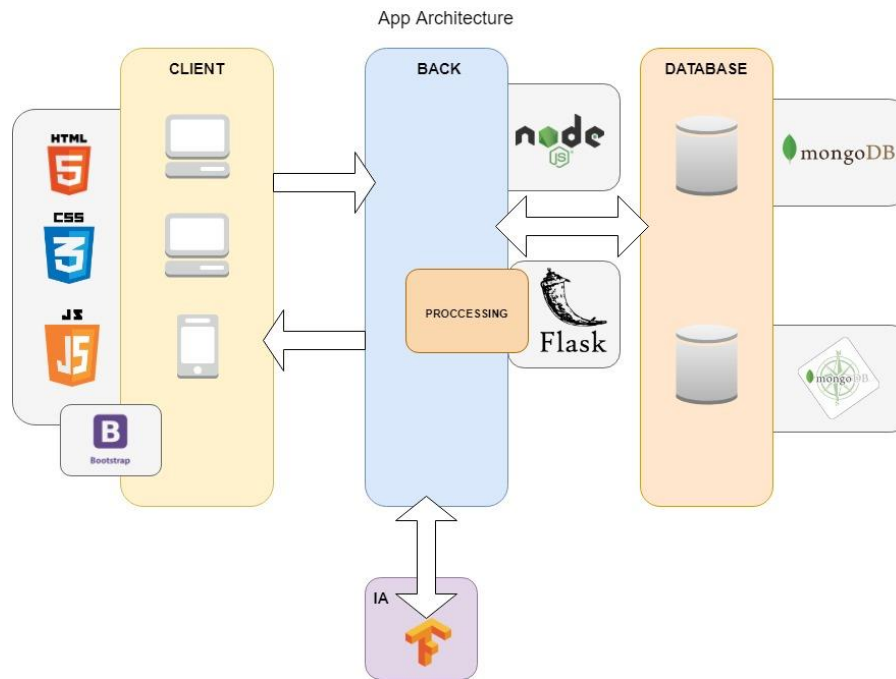


Figura 31. Arquitectura final con el módulo de IA

Aquí es donde las palabras “Mastering Adaptativo” cobran sentido, y es que gracias a la aplicación de redes neuronales, seremos capaces de adaptar una serie de parámetros de mástering a un archivo, gracias a la utilización de la plataforma por parte de los usuarios y al *feedback* recogido por cada uno de ellos, solucionando así varios de los problemas planteados como son la relevancia del usuario en el proceso y la optimización del mismo. Destacamos que para la realización de todo el proceso, es primordial contar con una buena base de datos previamente etiquetados que habrá que recolectar manualmente para que nuestro modelo pueda funcionar por sí mismo en un futuro, facilitando pasar de un entrenamiento supervisado a uno no supervisado.

CAPITULO 5: CONCLUSIONES Y TRABAJO FUTURO

Tras la consecución de este trabajo final de grado, hemos extraído una serie de conclusiones que nos han servido como experiencia para el desarrollo de futuras soluciones.

En nuestro caso hemos aprendido a analizar un problema surgido a la hora de mezclar música y tecnología, localizando el foco principal y planteando una solución a través de las herramientas que más se adaptaban a nuestra visión particular del problema.

Hemos comprobado la gran dificultad que conlleva plantearse cada una de las fases del desarrollo de un proyecto, desde el análisis de los requisitos hasta la consecución del mismo, ya que dentro de nuestra solución convergen muchas tecnologías específicas que necesitan ser ubicadas dentro del marco general del proyecto, obteniendo el valor añadido que nos produce cada una de ellas e intentando maximizar los resultados a través del planteamiento de un diseño minucioso.

La visión que aporta haber sido productor de sonido durante varios años se da de la mano con todos los conocimientos adquiridos durante la carrera, permitiéndonos tener las ideas más claras sobre cuáles podrían ser a priori los puntos más débiles de nuestro proyecto e intentando impulsar aquellos puntos fuertes para ganar visibilidad.

Una de las conclusiones técnicas más importantes que hemos inferido, es que para poder validar que el trabajo que realizamos se está ajustando a la realidad y es efectivo necesitamos que sea accesible a la mayor cantidad de gente posible. Gracias al aporte individual de cada una de las personas que realicen procesos de mastering se conseguirá mejorar los modelos y los algoritmos planteados durante todo el trabajo, por lo que será imprescindible para conseguir el resultado final deseado que una buena base de productores trabaje con nosotros durante las primeras etapas del funcionamiento de la plataforma.

5.1 Trabajo Futuro

Dentro del trabajo futuro todavía nos queda un largo camino por recorrer, ya que la implementación de esta aplicación se ha realizado completamente en ámbito local, siendo nosotros mismos los que hemos elaborado manualmente el dataset y los archivos musicales de prueba.

El *roadmap* para las siguientes iteraciones debería tener por objetivos los siguientes puntos, que si bien no son imprescindibles para la elaboración de la plataforma sí que lo serán para su correcto funcionamiento:

- Trasladar la solución que consta del servidor Back a un servicio en línea, que nos proporcione almacenamiento en sus servidores y un dominio accesible vía internet. Por ejemplo, Amazon Web Services, Azure y Google Cloud, todos ellos pueden proporcionarnos unos servicios a un coste económico.
- Recopilar una cantidad enorme de procesos de mastering, cuantos más datos tengamos para su análisis antes podremos realizar cambios sobre los algoritmos para su mejora.
- Crear un registro de usuarios, para adaptar aún más si cabe la experiencia a cada usuario.
- Crear una red social entorno a la plataforma. Debido a la necesidad de recolectar tantos datos como sea posible, necesitamos alcanzar una difusión elevada, por ello la inclusión de diferentes elementos como *Twitter*, *Facebook* o *Soundcloud* pueden ser imprescindibles si queremos alcanzar nuestro objetivo.
- Llevar todos los datos que no están siendo refinados a un *Data lake* para su almacenamiento en bruto, evitándonos el problema de hacer demasiado pesadas nuestras bases de datos principales y no perdiendo así el histórico de información recogida.
- Crear nuevos elementos de procesado más allá de los tres utilizados, como pueden ser excitadores armónicos, compresores multibanda, ensalzadores estéreo, normalizadores o maximizadores.
- Habilitar una plataforma de servicio técnico, en la que los usuarios puedan ponerse en contacto con nosotros para enviarnos sugerencias, errores encontrados o simplemente para resolver alguna clase de duda

- Monetizar la página, ya sea mediante la inclusión de publicidad o gestionando parte de los datos recolectados por la aplicación. Si somos capaces de crear ese registro de usuarios, podremos asociar toda la información extraída por sus archivos de música para obtener conclusiones como preferencias musicales, géneros preferidos, etc.

CAPÍTULO 6: GESTIÓN DEL PROYECTO

En la gestión de nuestro proyecto encontramos dos partes diferenciadas: entorno socio-económico y su marco regulador. En este apartado, explicaremos la planificación seguida así como el presupuesto resultante y su impacto en el sector. También analizaremos el marco regulatorio que le corresponde, teniendo en cuenta las últimas actualizaciones en lo que se refiere a materia legal.

6.1 Planificación del proyecto

Para poder obtener un presupuesto lo más cercano a la realidad posible, primero debemos exponer cuál ha sido la planificación del proyecto y el tiempo dedicado. Por tanto, se adjunta en la Figura 32 el diagrama de Gantt con cada una de las actividades que se plantearon y el tiempo que teníamos previsto para su consecución.



Figura 32. Diagrama Gantt del proyecto

Así, podemos observar como el tiempo requerido para la investigación ha sido junto con el del desarrollo los que más han ocupado el diagrama. Durante la etapa inicial del proyecto (meses de enero a abril) se investigó el problema y se esbozó una posible solución. También se investigaron las tecnologías necesarias para satisfacer todos los requisitos que se planteaban a medida que definíamos cada uno de los módulos.

La parte del desarrollo (meses de mayo hasta agosto/septiembre) fue más compleja, teniendo que invertir un número de horas superior.

Gracias a la planificación de las tareas desarrollada en mayo, se pudo empezar con los desarrollos en junio y concluirlos en unos 4 meses, cumpliendo con los objetivos establecidos en el gráfico.

6.2 Presupuesto

El trabajo que planteamos en el apartado anterior, fue desarrollado en su totalidad por una única persona, a la que supondremos de media un trabajo de lunes a viernes con una dedicación de entorno a una hora y media por día. Para establecer cuál es el precio por hora que esta persona facturará, utilizaremos un parámetro extraído de una consultora tecnológica, en nuestro caso serán 20 euros por hora.

También tendremos que tener en cuenta el aporte de gestión con el tutor de este proyecto, al que asignaremos unas 30 horas de dedicación, con un precio más elevado. Incluiremos también el precio del ordenador de sobremesa en el que se ha desarrollado todo, pero obviaremos otros como el coste de internet o la electricidad debido a que al ser elementos de uso compartido no podríamos extraer con exactitud las cifras, obviando también el precio del software ya que hemos trabajado en todo momento con software de código abierto.

Así, obtendremos la siguiente tabla de presupuestos:

Plataforma web de mastering	semanas	horas	coste
Junior	36	300	20€/h
Tutor	36	30	40€/h
Pc sobremesa	-		600€
Total:			7800€

Figura 33. Presupuesto del proyecto

Como podemos observar en la Figura 33, el precio total es de 7800€, y su mayor parte corresponde con la persona que se ha encargado de desarrollar toda la aplicación, suponiendo un 76% del presupuesto total.

Al ser un proyecto que introduce tantas tecnologías, ha sido necesaria una gran cantidad de tiempo para la investigación de cada una de ellas, por lo que si en lugar de utilizar a una única personas hubiesen sido varios profesionales con conocimientos en el sector, si bien los costes por hora serían mayores las horas que habría que haber dedicado hubiesen descendido considerablemente, obteniendo un resultado total menor y abaratando el coste total.

6.3 Impacto socio-económico

En este apartado, analizaremos el impacto en el sector musical que podría llevar la puesta en marcha de nuestra aplicación, desde cómo afectaría a las empresas que se dedican profesionalmente al mastering musical hasta el posible beneficio que aportaría a la comunidad de productores *amateurs*.

Con respecto a las empresas de mastering, hoy en día la mayoría de estudios musicales ofrecen este tipo de servicios, obteniendo un rango diferente de precios dependiendo de la calidad del ingeniero de sonido que se encargue de realizar el proceso, del prestigio del estudio, de la cantidad de archivos que se masterizan, etc. Cuando observamos páginas de estudios profesionales vemos que aproximadamente estos procesos suelen tener un coste desde 40€ por un tema musical de un artista poco conocido, hasta los miles de euros de las grandes bandas musicales.

Si bien nuestra aplicación no pretende sustituir este tipo de servicios, sí que puede actuar como sustitutivo en aquellos músicos que no busquen una calidad perfecta de sonido y que quieran distribuir más económicamente su música. Cuando antes un músico solo tenía la opción de acudir a un estudio por un elevado precio o realizarlo él mismo, hoy encuentra una herramienta gratuita que puede en gran medida satisfacer sus expectativas. Si su experiencia con esta clase de servicios es positiva, lo más probable es que acuda a nuestro tipo de servicios debido a que le supondría un ahorro del 100% del presupuesto que en un principio tenía asignado a esta parte del proceso, a su vez fomentaría su uso y extensión, dando acceso a más personas a través de las redes sociales.

Los grandes actores perjudicados podríamos encontrarlos en las plataformas de mastering que analizamos en el capítulo 2.4, aquellas empresas que se dedican a ofrecer

lo mismo que nosotros pero que utilizan una plataforma de pagos con unos precios bajos. Si cumpliésemos nuestro objetivo de ser una plataforma gratuita que se financia a través de la publicidad, la diferencia de precio haría que realizando el mismo trabajo la mayoría de usuarios prefiriesen utilizar nuestra aplicación, ya que el riesgo de invertir algo de dinero en una plataforma de la que no puedes conocer a ciencia cierta si el proceso de mastering será realizado correctamente es bastante más elevado.

Por tanto habría dos grandes beneficiados en el sector, el primero de ellos serían todos aquellos productores de sonido o bandas musicales *amateur* que se verían provistos de una herramienta que les permite alcanzar una calidad que hasta ahora solo era posible mediante la contratación de los servicios de mastering. También sería un punto a favor de aquellos grupos algo más profesionales que decidiesen comprobar como suena una idea prácticamente masterizada sin la necesidad de acudir a un profesional, y en caso de obtener un buen resultado acudir a estudios más grandes que acabasen el proceso.

La última de las grandes beneficiadas, sería la música. Si bien es algo que no podemos medir, es incuestionable que facilitar herramientas gratuitas para que los productores que deseen obtener unos mejores resultados lo consigan, alimentará el sector con la introducción de cada vez más personas interesadas por crear contenido musical.

6.4 Marco Regulator

Cuando hablamos del sector musical sin duda alguna tenemos que hacer mención a uno de los problemas principales en lo que se refiere al marco regulator del sector musical, la propiedad intelectual.

Nuestro desarrollo será potencialmente usado por nuevos artistas que subirán contenido musical a nuestra plataforma, la mayoría de las ocasiones serán obras que aún se encuentran en proceso de creación (hay que tener en cuenta que el mastering es el último paso de la cadena musical por lo que el tema musical no estaría completo), por lo que habrá que lidiar con diferentes problemas. Para responderlos, plantearemos una serie de preguntas en las que se verán reflejados los principales temas legales que podrían afectarnos, los solucionaremos apoyándonos en el reglamento actual de la Sociedad General de Autores y Editores.

1-¿A quién pertenecen la autoría o derechos del tema musical si nuestra plataforma participa en el proceso de creación del mismo?

La autoría o derechos de la obra musical pertenecerán completamente a su autor. Nuestra plataforma es una herramienta más del conjunto total que utilizará su autor para la elaboración de su obra musical, si bien se puede inducir al error debido a que nuestro servicio es gratuito y puede ser entendido como una forma de colaboración (cuya co-autoría si estaría recogida en la SGAE) nos valemos de otros elementos como la publicidad para obtener un resultado económico.

2-¿Tendrá que pagar la plataforma web alguna clase de impuesto?

No. Los archivos de audio que se suben a la plataforma aún no han sido registrados como obras sujetas a restricciones de propiedad intelectual en la SGAE, por lo que no se debería aplicar ninguna clase de canon o impuesto.

3-¿Debería existir alguna clase de elemento que prevenga la subida de archivos con contenido de obras musicales pertenecientes a la SGAE?

No. Al no tratarse nuestra plataforma de un medio de distribución, no sería necesario implementar ninguna clase de algoritmo que previniese de la subida de esta clase de archivos, ya que se utilizarían siempre para uso privado por parte del usuario.

4-¿Qué ocurre legalmente con los datos almacenados en nuestras bases de datos, los cuales serán utilizados para el funcionamiento de la plataforma?

Nosotros no almacenamos los temas musicales, solo los procesamos. La extracción de las características musicales de un tema musical solo puede darnos una ligera idea de su sonido final, pero en ningún caso podremos replicarlo únicamente con estos datos, por lo que no estaremos incumpliendo ningún reglamento o normativa.

5-¿En caso de habilitar el registro de usuarios y recoger información sobre estos, hasta qué punto podríamos venderla a terceros?

Dentro de nuestra aplicación, deberíamos desarrollar una serie de vistas en las que recojamos el consentimiento del usuario, asegurándonos que cumplimos con el reglamento que establece la nueva GDPR (General Data Protection Regulation) y que

entró en vigor este mismo año. Los principales puntos que deberemos cumplir serán los siguientes:

- Derecho a la información: Hay que informar de manera clara y concisa de qué se realizará con los datos obtenidos del usuario.
- Derecho de rectificación: Es el “derecho al olvido”, el usuario tendrá derecho a la eliminación total o parcial de sus datos.
- Derecho portabilidad: El usuario tendrá derecho a que se le entreguen todos los datos propios que hayan sido recogidos, en un formato estructurado y entendible a primera vista.
- Derecho de objeción: El usuario podrá oponerse en todo momento a que se utilicen sus datos personales.

El incumplimiento de esta normativa conllevará sanciones económicas muy elevadas, ya que uno de los principales objetivos marcados por la Unión Europea es el avance en todo lo referente a protección de datos.

Así, daremos cumplimiento a los dos reglamentos que más afectarán a nuestro proyecto, ya sea el reglamento de la propiedad intelectual en España o el reglamento que toda plataforma web que pueda manejar datos sensibles debe implementar.

Si bien existen otros marcos legales, cumpliendo con estas normativas seremos capaces de acceder a diversas estrategias de financiación que nos permitirán seguir ofreciendo todo nuestro servicio de forma gratuita.

ENGLISH APPENDIX

Sometime ago, the creation of musical tracks required an economic investment that most of bands or music groups could not afford. The tools needed to produce a musical track were so big and took all the room space because you needed huge analogue elements and microphones for the signals, and also engineers with extensive knowledge in the subject to cover the technical aspects that would allow you to sell the song as a product.

Today, thanks to the rise of new technologies and digital transformation, a new figure has emerged within the music industry: the independent producer.

The cheapening of the tools and their digitization have promoted the creation of home-studios at a much lower cost than the budgets of professional studios, eliminating the entry barriers to the sector for these producers who are capable of creating musical themes with a professional quality for a small part of the price that would entail doing it in a large studio.

However, most of these producers are nourished by information obtained through various blogs on the web, which often expose ideas that are wrong or that do not ensure that their implementation results in obtaining a perfect audio quality. Producing a musical theme is complex, and if we combine the small experience that they have with the misinformation of the web, we can see how in the most technical parts of the chain of creating a musical theme, these producers fail in the tools apply and in the processes necessary to get a professional sound.

In this project, we will talk about one of the parts that are included in the creation of a musical theme, the mastering, and the creation of a web platform that allows us to automate the whole process and apply Big Data and Machine learning techniques to create an adaptive algorithm.

When we talk about mastering, we talk about the last step in the elaboration of a professional musical track, the place where the errors produced in the mix will be corrected and where the audio will be treated to obtain a professional quality sound that can be distributed later in any format.

Although mastering has an artistic component, most processes usually follow a pattern based on equalization, compression and limitation, the last one being very important due to psychoacoustic phenomena of sound quality perception. Thanks to this pattern, we can automate the process and create solutions that adapt any input file.

The difficulty involved in performing a mastering correctly is very high, that is the reason why in the past, the companies that made the mastering process were the big studios that had sound engineers with years of experience, who charged a very high price for each masterized track.

That difficulty has affected amateurs producers when producing their own themes, because although they have the necessary tools to obtain the professional quality that mastering provides, they don't have enough knowledge to apply it, or they don't have the necessary budget to make another person do it for them.

Our main objective will be facilitate that those producers can acquire a professional sound, or at least, that they can get an idea of how their tracks would sound if they were mastered by a professional studio, since our application is not meant to be a substitute of the current mastering processes in the first stage, and will be with the time, with the collection of better data and with the refinement of its models when it will be possible to approximate our processing to this kind of masterings.

Although there are similar solutions in the market to those that we have proposed, our project will add two layers that will act as a differential factor, a layer of processing large volumes of data and an artificial intelligence layer, nourished by a set of data that we will collect from the user's feedback. We will also offer our service for free, although we will collect data from users to improve our model, so we will analyze the possible impact of the new GDPR regulations and the possible legal consequences that would imply breaching the regulation of Spanish authors society.

Thanks to the research carried out, we will analyze all the parts that will compose our architecture by dividing the entire structure into different layers, which will include different modules such as:

- **UI:** we will analyze the requirements and design a fully navigable web page in which you can upload a file for your mastering.
- **Back-end:** we will analyze the requirements and design the logic of a server that connects all the modules through a routing with different endpoints.
- **Database:** we will analyze the requirements and design a non-relational database capable of storing all the features, parameters, processes and any kind of data that is necessary for the correct work of our application and subsequent data analytics processes.
- **Big Data:** We will analyze the requirements and design a module that is capable of processing the large amount of data available through algorithms and obtain the optimal parameters for the realization of the mastering. At the same time we can visualize the results of each of the processes in real time, analyzing the wrong processes to draw conclusions that allow us to improve the general performance.
- **Process:** we will implement equalizers, compressors and limiters for the processing of the signal, creating a REST service for the easy access of each of them. We will design this module in a micro service format, which allows us to create our own chains to create dynamic mastering chains, and also to choose the elements necessary for its realization.
- **AI:** Thanks to the large set of data labeled, we will design an artificial intelligence model based on the successful processes collected by our application, replacing the big data module and optimizing the entire process. We will train a neural network through a back propagation algorithm, which will allow us to create a base to iterate, which will learn from the new mastering processes.

In this project we extract several important conclusions for being able to adapt the work that will have to be done in the future. The first one is that the amount of data that we will handle will be so big. Although we will have a test dataset that will allow us to

perform the first iterations, will be the feedback collection from a lot of users and the use of the service the tool that will provide us the necessary elements to achieve the main goals.

The quality of these data will be so important, since using a set of data labeled according to their success flag will be a key feature, in order to subsequently being able to learn from those processes that have been successful and discard those ones that have not been relevant.

The large number of technologies used is a big challenge that has been overcome, connecting different modules with different specific use, and designing a multichannel solution and integration.

The algorithms that process all the track features in the Big Data layer must be powerful enough to make calculations in several dimensions and obtain a total processing time good enough for keeping the user experience.

Once we have created the artificial intelligence model, we will replace the Big Data layer with this model, which will provide us with shorter waiting times and greater efficiency due to the ability to adapt to any kind of input track.

The future work planned involves the development of some solutions that would bring us closer to the goal of creating a perfect platform, and among them we can find:

- Create new processing elements such as multiband compressors, stereo enhancers, etc.
- Create a user registry that allows us to obtain data from them and customize the entire experience.
- Make our platform accessible via the web to get as many users as possible for improving the quality.
- Create a social network around the platform.
- Enable a technical service platform, creating an email and all the necessary methods.

- Monetize the platform based on advertising, we will still offer our platform as a “free to use”, but we have to implement some business logic for maintaining the servers and other elements.

The project has been developed in 9 months, we can find the budget that entails his realization and also each one of the stages that we designed through a Gantt Diagram. We will also mention the socio-economic impact that the implementation of our application would have and how it would affect direct competitors, big actors, the biggest music studios and the music sector in general.

As the final conclusion, this project is oriented for working as a proof of concept, showing how the things should be done, but it would be necessary the implementation of that “future work” for providing it with a full data stack which let us improve the performance and the quality of the platform. We will not replace the actual studio mastering because the technology is not good enough for doing it at this moment, but with the upcoming innovation ideas it’s a matter of time.

GLOSARIO

Mastering: último paso de la cadena de creación de un tema musical, en él se corrigen los posibles errores del archivo, se modifica para que suene de forma profesional y se adecúa para ser distribuido en cualquier formato.

Home studio: Pequeño estudio de música que una persona construye en su propia casa.

Secuenciador Multipista digital: Herramienta de producción de audio.

Plugin: Referido al sector de la edición del audio, es una herramienta digital para el tratamiento del audio que puede ser cargada de forma externa en un secuenciador.

One-Page: Formato de página web en el que solo se expone una única página con diversos apartados.

Script: Secuencia de código con una determinada función.

Feedback: En el proyecto, opinión de los usuarios de toda la experiencia.

Core: núcleo de nuestra aplicación.

Machine learning: Aprendizaje automático, subcampo de la inteligencia artificial.

all-in-one: Servicio único en el que se incluyen más servicios independientes

Headroom: Nivel entre el pico más alto de un tema musical y el límite digital de 0dBs.

Dataset: Conjunto de datos.

Endpoint: Ruta de acceso a un determinado servicio.

Track: Tema musical.

Routing: Conjunto de endpoints o direcciones que componen la parte Back-end del servidor.

Framework: Estructura compuesta por módulos software de una determinada tecnología para facilitar su uso y desarrollo.

Base64: Sistema de numeración que como base utiliza el 64.

Stream: Flujo de datos.

Parsear: Transformación de texto plano en una estructura de datos.

Timestamp: Marca de tiempo que hace referencia a un momento determinado.

SIGLAS, ACRÓNIMOS Y ABREVIATURAS

JSON: Formato de datos estructurados, notación de objetos *javascript*.

RMS: Valor eficaz, media de la potencia sonora de un tema musical.

dB: decibelios.

API: Conjunto de rutas que exponen una serie de servicios web.

SGAE: Sociedad General de Autores y Editores.

GDPR: Reglamento general de protección de datos.

AI: Artificial Intelligence

BIBLIOGRAFÍA

[1] Portal Wikipedia, [En línea].

Disponible en:

<https://es.wikipedia.org>

Último acceso: septiembre 2018

[2] Bob Katz, 2002, *Mastering audio, the art and the science*.

[3] El Pais, *La difícil supervivencia de los grandes estudios de grabación* [En línea].

Disponible en:

https://elpais.com/cultura/2010/08/13/actualidad/1281650402_850215.html

Último acceso: septiembre 2018

[4] El Pais, *Hago discos en mi casa* [En línea].

Disponible en:

https://elpais.com/diario/2009/02/16/madrid/1234787064_850215.html

Último acceso: septiembre 2018

[5] Página web de Waves, [En línea].

Disponible en:

<https://www.waves.com/limiting-tips-for-mastering>

Último acceso: septiembre 2018

[6] Notodo.com, *La guerra del volumen* [En línea].

Disponible en:

<http://www.notodo.com/la-guerra-del-volumen>

Último acceso: septiembre 2018

[7] Thomas H.Davenport, Paul Barth and Randy Bean, 2102 ,*How “Big data” is different* [En línea].

Disponible en:

<https://pdfs.semanticscholar.org/eb3d/ece257cca2e8ce6eaf73fd98c1fdcdbc5522.pdf>

Último acceso: septiembre 2018

[8] Worldwide Big Data Technology and Services 2014-2018 Forecast (IDC)

[9] Página web de LANDR, [En línea].

Disponible en:

<https://www.landr.com/es>

Último acceso: septiembre 2018

[10] Microsoft startup partner of the year awards, [En línea].

Disponible en:

<https://blog.bandlab.com/bandlab-wins-microsoft-start-up-partner-of-the-year-award-2016/>

Último acceso: septiembre 2018

[11] Página web de Emastered, [En línea].

Disponible en:

<https://emastered.com>

Último acceso: septiembre 2018

[12] web de W3schools, [En línea].

Disponible en:

<https://www.w3schools.com/>

Último acceso: septiembre 2018

[13] StackOverflow developers survey, [En línea].

Disponible en:

<https://insights.stackoverflow.com/survey/2017>

Último acceso: septiembre 2018

[14] Node user survey Report, [En línea].

Disponible en:

<https://nodejs.org/en/user-survey-report/>

Último acceso: septiembre 2018

[15] Página web de Express, [En línea].

Disponible en:

<https://expressjs.com/es/>

Último acceso: septiembre 2018

[16] Página web de Mongoose, [En línea].

Disponible en:

<https://mongoosejs.com/>

Último acceso: septiembre 2018

[17] Página web de MongoDB, [En línea].

Disponible en:

<https://www.mongodb.com/>

Último acceso: septiembre 2018

[18] Página web de Flask, [En línea].

Disponible en:

<http://flask.pocoo.org/>

Último acceso: septiembre 2018

[19] Página web de Apache Spark, [En línea].

Disponible en:

<http://spark.apache.org/>

Último acceso: septiembre 2018

[20] Greg Stanley, 2017, *Real Time Moving Average Clustering*, [En línea].

Disponible en:

<https://gregstanleyandassociates.com/whitepapers/BDAC/Clustering/clustering.htm>

Último acceso: septiembre 2018

[21] Página web de MongoDB, [En línea].

Disponible en:

<https://www.mongodb.com/products/compass>

Último acceso: septiembre 2018

